



Allen-Bradley

EtherNet/IP Web Server Module

1756-EWEB

User Manual

**Rockwell
Automation**

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. *Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls* (Publication SGI-1.1 available from your local Rockwell Automation sales office or online at <http://www.ab.com/manuals/gi>) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc. is prohibited.

Throughout this manual we use notes to make you aware of safety considerations.

WARNING



Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

ATTENTION



Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:

- identify a hazard
 - avoid a hazard
 - recognize the consequence
-

SHOCK HAZARD



Labels may be located on or inside the drive to alert people that dangerous voltage may be present.

BURN HAZARD



Labels may be located on or inside the drive to alert people that surfaces may be dangerous temperatures.

Getting Started

How to Use This Quick Start

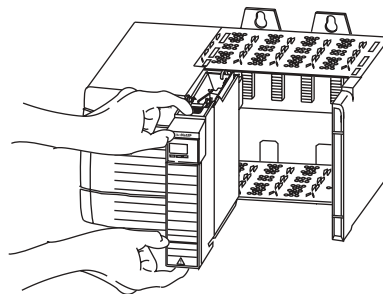
This quick start summarizes the steps you follow to get a web server module to operate on a Ethernet network. This quick start also provides an overview of some the main tasks you can accomplish using the web server module.

For this information:	See page:
Installing and Connecting the Web Server Module to the Network	1
Using the Web Server Module	4

Installing and Connecting the Web Server Module to the Network

- 1. Install the web server module** Install the module in a 1756 chassis.

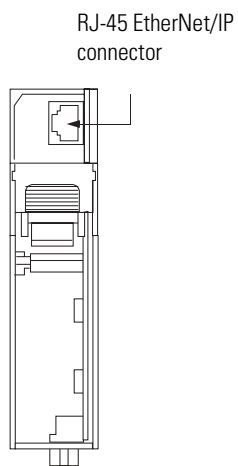
For more information, see the EtherNet/IP Web Server Module Installation Instructions, publication 1756-IN588.



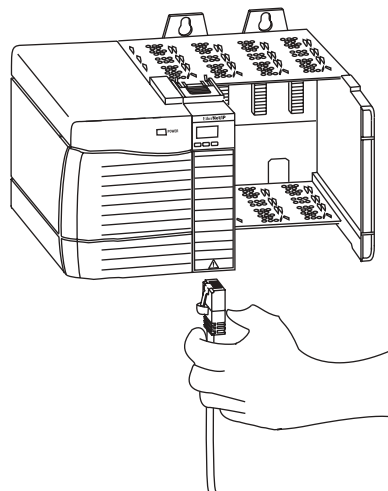
2. Connect the module to the network

For more information, see the EtherNet/IP Web Server Module Installation Instructions, publication 1756-IN588.

Connect the web server module to the network. The RJ-45 connector is on the bottom, front of the module.



This is the bottom, back of the module that connects into the chassis.



Connect the cable here.

3. Obtain an IP address

For more information, see chapter 2

By default, the web server module is DHCP enabled. If you connect the web server module to a network that has a DHCP server, that server will assign a dynamic IP address to the web server module and the four-digit display on the front of the web server module will display each of the four numbers of the IP address.

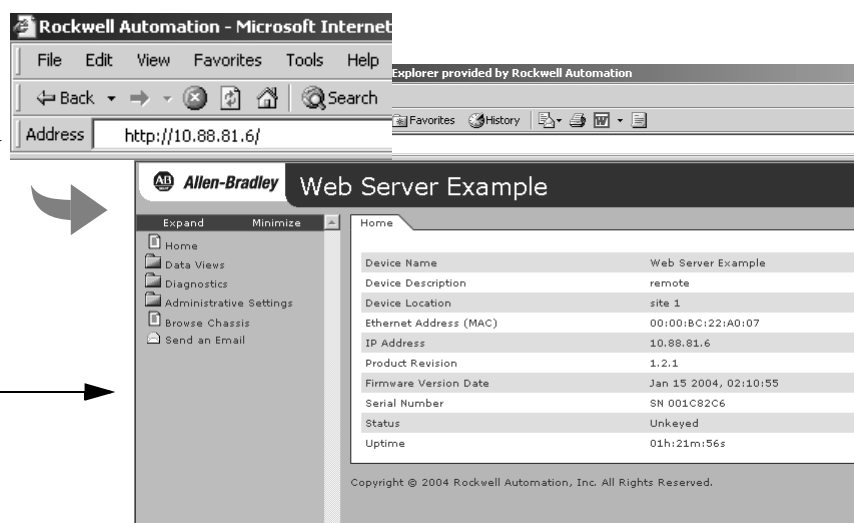
If your network does not have a DHCP server, use one of the methods described in chapter 2 to assign an IP address to the web server module.

4. Access the Home page of the module

From your web browser, enter the IP address of the web server module. The module displays its Home page

Specify the IP address of the web server module in the Address window of your web browser.

This is the module's Home page.



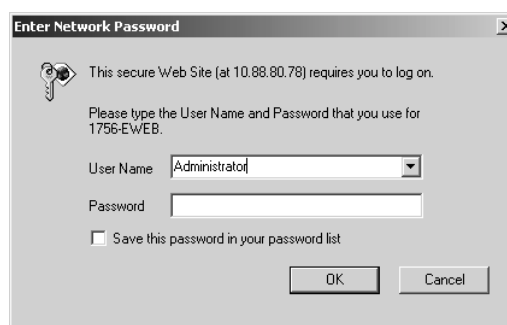
5. Log into the module

Many of the features of the web server module require you to log in with appropriate access. If you select a feature, such as New Data View, the web server module prompts you to enter your user name and password. The default user name is "Administrator" with no password (leave the Password field blank).

Default Access:

User Name: Administrator
(not case sensitive)

Password:
(leave blank, no password)

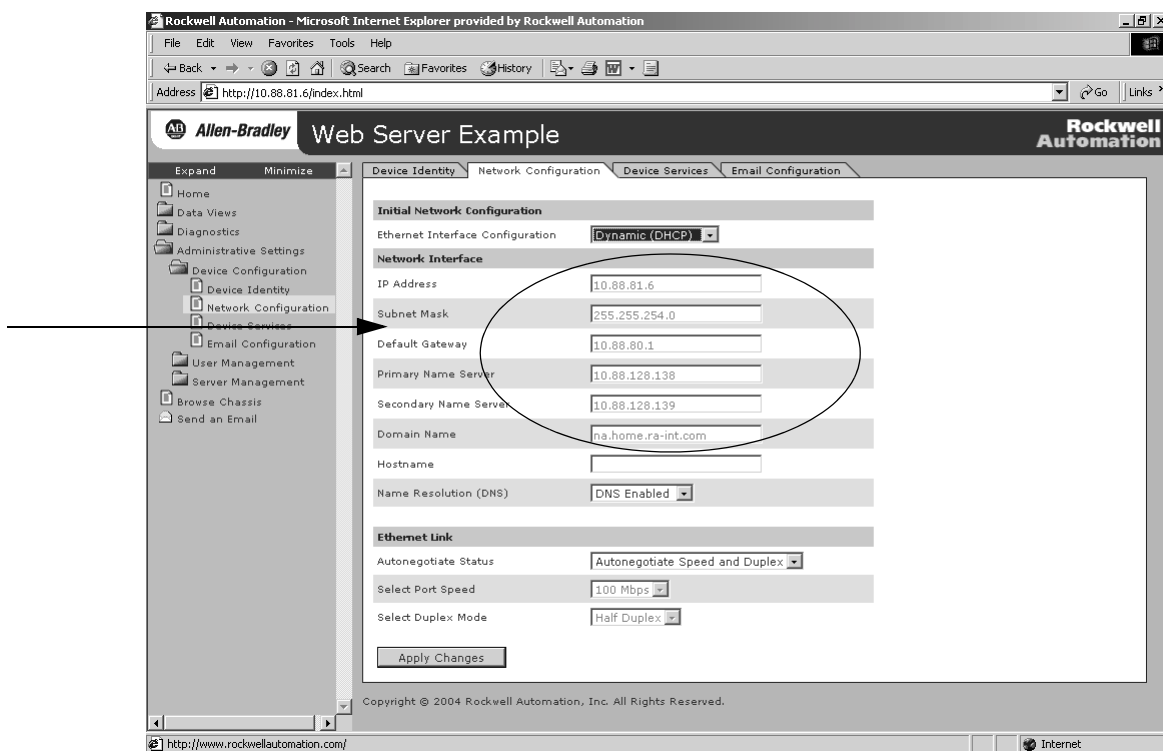


You can set up as many as 25 user accounts. Each account can have read, read and write, or administrator access. For more information, see chapter 6.

6. Confirm the network configuration

For more information, see chapter 2

On the Administrative Settings → Device Configuration → Network Configuration page, you can verify the IP address and other network settings.



Using the Web Server Module

To help familiarize yourself with the web server module, some of the tasks you can accomplish with the web server module include:

If you want to:	See page:
Create a data view	5
Access a data view	6
Configure email	7
Configure the time server	8
Enable/disable other services	9

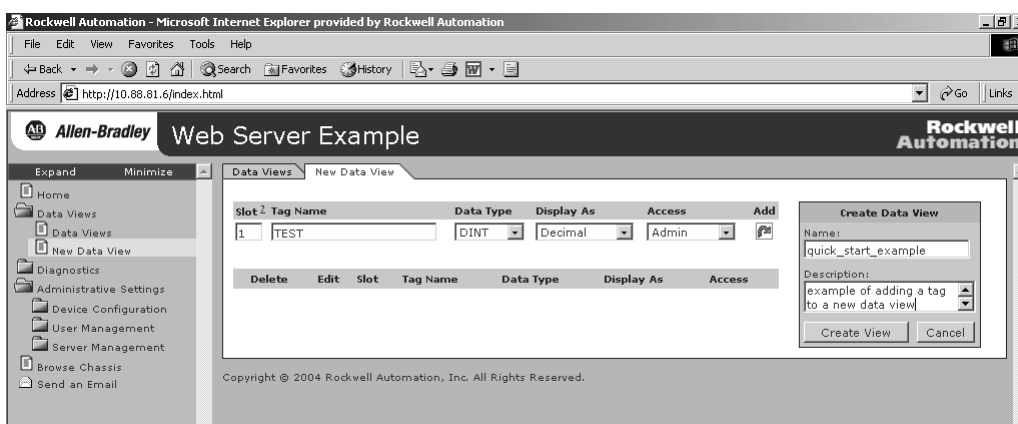
Create a data view


For more information, see chapter 4

Before you can create a data view in the web server, the tags you want to view must exist in the controller that is local (in the same chassis) to the web server module. The tags in the controller must be controller-scoped. For example, create:

```
TEST
type DINT
controller-scope
value 12345
```

To create a data view, you need Administrator or Write access. You create a data view from the Data Views → New Data View page.



1. In the Create Data View window, specify a name for the data view and enter a description (optional).
2. In the New Data View window, specify at least one tag (EWEB_test in this example). You must specify the:
 - slot number of the controller
 - tag name (case sensitive; must be exactly as it is specified in the controller)
 - type of tag
 - how to display the tag data
 - access limit of the data view
3. Click on the Add  button to add the tag to the data view.

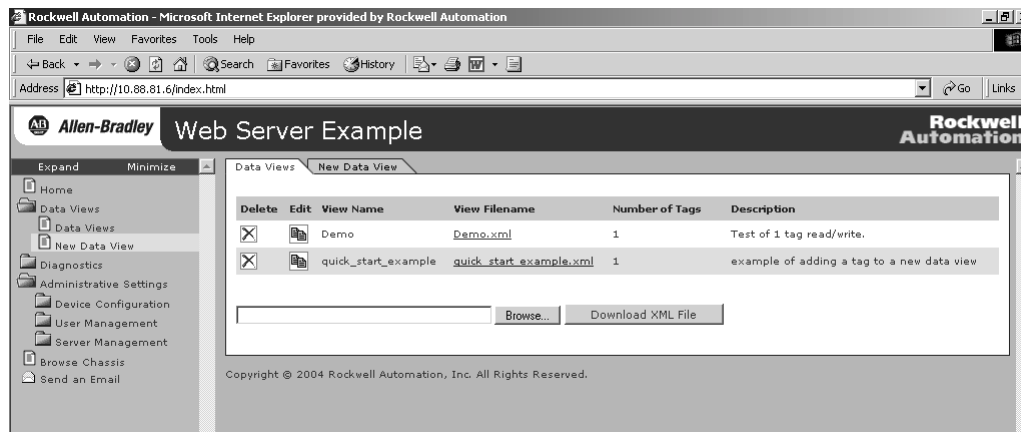
Continue adding as many tags as you want to configure.

4. Click Create View

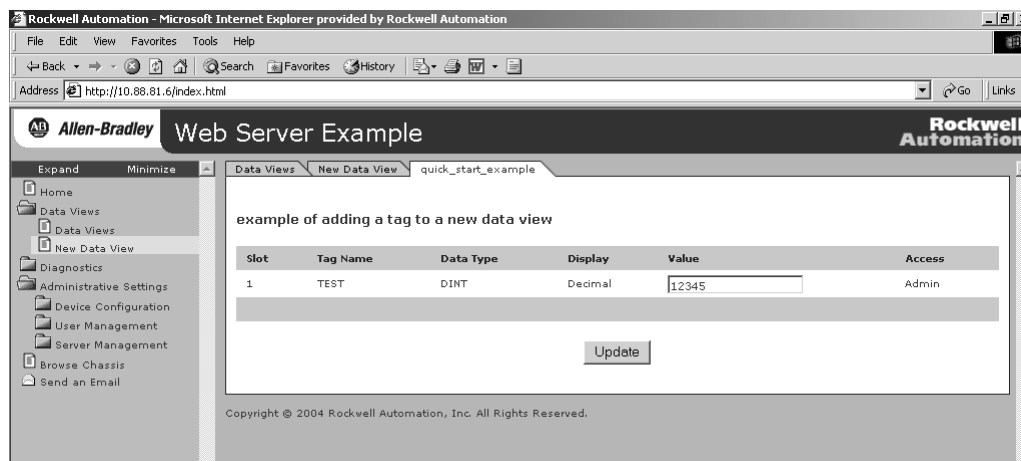
Access a data view

For more information, see chapter 4

From the Data Views → Data View page, select the data view you just created.



Click on the filename link to view the tags in this data view.

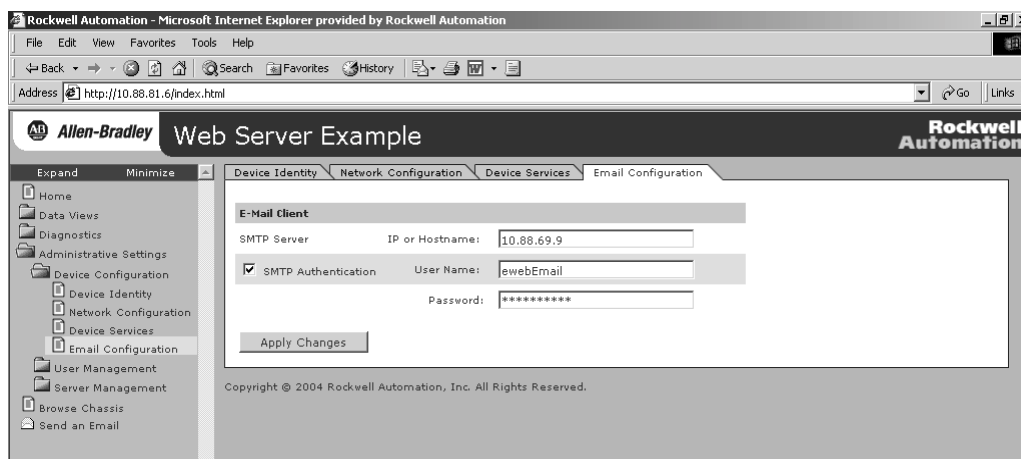


To change a data value, you need Administrator or Write access. Type the new value in the box next to the tag and click the Update button. This changes the value in the controller. You can use RSLogix 5000 programming software to verify that the value changed.

Configure email

For more information, see chapter 5

You configure the SMTP server that manages email on the Administrative Settings → Device Configuration → Email Configuration page.

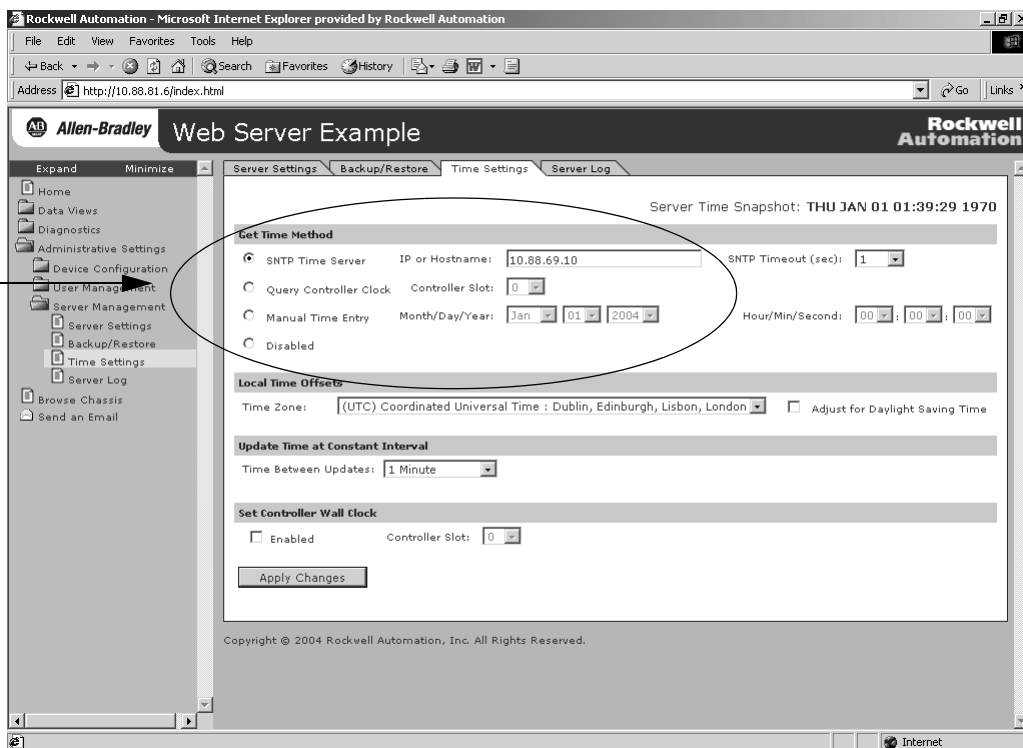


You can enter and send an email from the Send an Email link on the web server module. Or you can have a controller execute a MSG instruction that initiates email through the web server module.

Configure the time server

For more information, see chapter 3

You select the method the web server module uses to maintain an accurate date and time stamp on the Administrative Settings → Server Management → Time Settings page. This makes sure that files you save to the web server module have accurate date and time stamps.



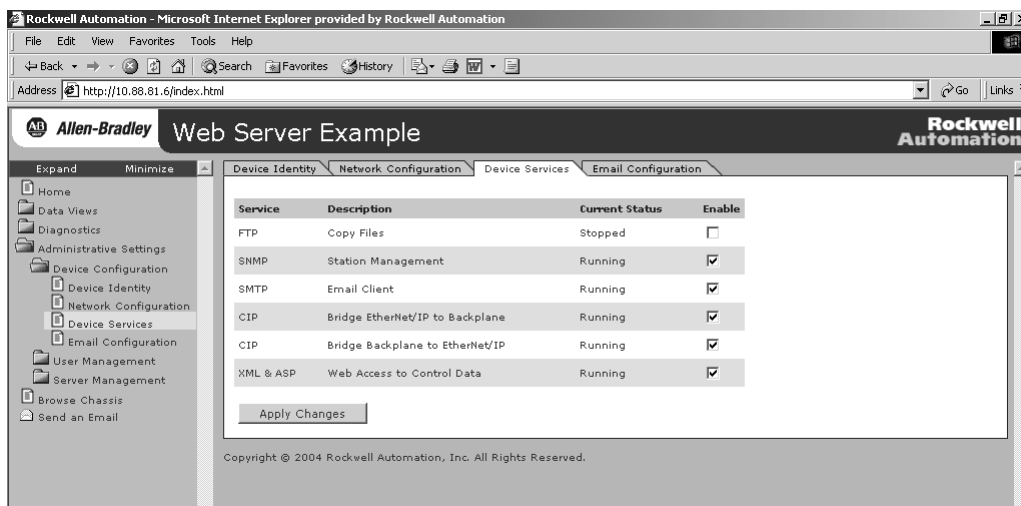
Select:

- SNTP Time Server to get the date and time from an SNTP server on the network.
- Query Controller to get the date and time from the local controller.
- Manual Time Entry to specify your own date and time.

Enable/disable other services

For more information, see chapter 3

You can enable other services from the Administrative Settings → Device Configuration → Device Services page.



Select the services you want to use. Enable the:

- FTP (File Transfer Protocol) service to allow file transfers to and from the web server module
- SNMP (Simple Network Management Protocol) service if your system uses SNMP management software
- SMTP (Simple Mail Transfer Protocol) to service email
- CIP (Common Industrial Protocol) Bridge Ethernet to Backplane service to allow EtherNet/IP devices to bridge through the web server module to devices in the chassis
- CIP (Common Industrial Protocol) Bridge Backplane to EtherNet/IP service to allow other devices in the chassis to bridge through the web server module to EtherNet/IP devices
- XML/ASP (Extended Markup Language/Active Server Page) service to allow web access to control system data

Notes:

About the Rockwell Automation Enhanced Web Server Modules	Chapter 1	
	How to Use This Chapter	1-1
	Typical Applications	1-1
	Using a Web Server Module in a Control System	1-2
	Browser Requirements	1-3
	Accessing the Home Page	1-4
	Navigating the Web Server Module	1-4
Network Addressing for a Web Server Module	Chapter 2	
	How to Use This Chapter	2-1
	Determining Which Network Parameters Are Required	2-1
	Assigning Network Parameters when the Network HAS a DHCP Server.	
	2-3	
	Assigning a static IP address.	
	2-4	
	Assigning Network Parameters when the Network DOES NOT HAVE a DHCP Server	
	2-5	
	Using the Rockwell Automation BOOTP/DHCP utility . .	
2-6		
Using RSLinx software to set the IP address		2-8
Using RSLogix 5000 software to set the IP address. . . .		2-9
DNS Addressing	2-10	
Verify Network Settings.	2-12	
Managing Module Settings	Chapter 3	
	How to Use This Chapter	3-1
	Managing Module Information.	3-1
	Define module-specific information for the Home page.	
	3-2	
	Modify network parameters	
	3-3	
	Enable and disable communication services	
3-4		
Managing Server Settings.	3-5	
Customize server settings		3-6
Configure the time server.		3-7
Display the server log		3-8

**Using Data Views to Access
Controller Data****Chapter 4**

How to Use This Chapter	4-1
Overview of Data Views	4-1
Tags supported in data views.	4-2
Performance estimates	4-2
Creating a Data View	4-3
Adding tags to a data view.	4-4
Monitoring Data Views and Tag Data.	4-5
Sorting data views	4-5
Interfacing with the Logix5000 controller	4-6
Editing a Data View	4-7
Creating Data Views Offline	4-8
Using an External Application to Access Data Views.	4-9
Read a data view via an external application.	4-9
Change data in a data view via an external application	4-9
Example: Data View XML	4-10
Example: Data View XML With Tag Values	4-11
Example: Data View XML With Tag Errors	4-12

Sending Email**Chapter 5**

Using This Chapter	5-1
Overview	5-1
Configuring the Web Server to Send Email.	5-3
Sending an Email via the Web Page.	5-4
Sending an Email via a Controller-Initiated Message Instruction. 5-4	
Step 1: Create string tags	5-5
Step 2: Enter the ladder logic	5-6
Step 3: Configure the MSG instruction.	5-7
Entering the text of the email	5-9
Possible email status codes.	5-10

**Managing User Accounts and
Access Levels****Chapter 6**

How to Use This Chapter	6-1
User Accounts and Privilege Classes.	6-1
Configuring Access Limits for Web Pages.	6-2
Creating User Accounts	6-5
Recovering with Unknown Password.	6-6

	Chapter 7	
Accessing Files in the Web Server Module	How to Use This Chapter	7-1
	Accessing the Web Server's File System	7-1
	Connecting to the web server module	7-2
	File names and types	7-3
	Backing Up the File System on the Web Server Module . . .	7-5
	Backing up files.	7-6
	Restoring files	7-7
	Chapter 8	
Creating Custom Web Pages	How to Use This Chapter	8-1
	Overview	8-1
	Accessing custom web pages	8-3
	Developing a Custom Web Page	8-3
	ASP Function Calls	8-4
	Read Controller Tags.	8-4
	Read CIP Data	8-5
	Updating control system data	8-7
	Retrieve Information about the Web Server Module	8-7
	Javascript Libraries	8-9
	Javascript library: conversion.js.	8-9
	Javascript library: XMLObjectLoaderLib.js	8-10
	Web Page Forms and POST Handlers	8-12
	ACTION="/user/system/dataviews/filename.xml".	8-13
	ACTION="/rokform/WriteLogixTags"	8-17
	ACTION="/rokform/ReadLogixTag"	8-19
	ACTION="/rokform/CIPMessage"	8-20
	Chapter 9	
Monitoring Diagnostics	How to Use This Chapter	9-1
	Web Server Module Diagnostics.	9-1
	Diagnostics Overview	9-2
	Network Settings.	9-4
	Message Connections	9-6
	Ethernet Statistics	9-7

	Appendix A	
Specifications	How to Use This Appendix	A-1
	1756-EWEB Specifications	A-1
	1756-EWEB LED Indicators	A-3
	Appendix B	
Connection Use by the Web Server Module Over Ethernet	How to Use This Chapter	B-1
	CIP Connections	B-2
	CIP connected messaging limits	B-2
	CIP unconnected messaging limits	B-3
	TCP Connections	B-3

About the Rockwell Automation Enhanced Web Server Modules

How to Use This Chapter

Rockwell Automation offers enhanced web server modules for your EtherNet/IP control systems so you can monitor and modify data remotely via XML web pages.

This chapter shows how you can use a web server module in your control system.

For this information:	See page:
Typical Applications	1-1
Using a Web Server Module in a Control System	1-2
Browser Requirements	1-3
Accessing the Home Page	1-4
Navigating the Web Server Module	1-4

The current offering of web server modules includes:

- 1756-EWEB module for ControlLogix systems

Typical Applications

The web server module provides access to controller data. This access opens up different, remote access applications to control systems. Use the web server module to:

- Remotely access controller data using a web browser

Use a standard web browser to monitor live controller data in two ways: use data views that you create in the web server module, or custom-develop your own web pages. For example, create a custom web page for managers to monitor production processes directly from their desks. Use data views and/or custom web pages for OEMs to remotely monitor controller data and reduce support costs.

- Deliver data initiated by the control system

System data and information can be sent via email when initiated by a controller in the system. The controller uses a message instruction to initiate an email. Use the email to notify a maintenance person or an engineer of an alarm or alert so that corrective actions can be done in a timely fashion. The system can also send system status or production reports. The web server module supports all email clients, such as email applications, text pagers, etc.

- Share system data with external applications

The web server module stores data in its data views in XML files. This generic XML data presentation allows external applications to easily access and manipulate system data. XML support is also platform and operating system neutral, so you can share data between different applications. For example, design a database application to obtain controller data from the web server module to streamline the data acquisition process.

Using a Web Server Module in a Control System

The web server module supports:

- bridging and routing of messages

Like other EtherNet/IP modules, you can route messages, upload/download programs, and flash upgrade modules using the web server module as part of the communication path to access the target device.

- data access (read and write) to ControlLogix controllers

Access the XML pages in the 1756-EWEB module to view and modify data that resides in any ControlLogix controller that is in the same chassis as the 1756-EWEB module.

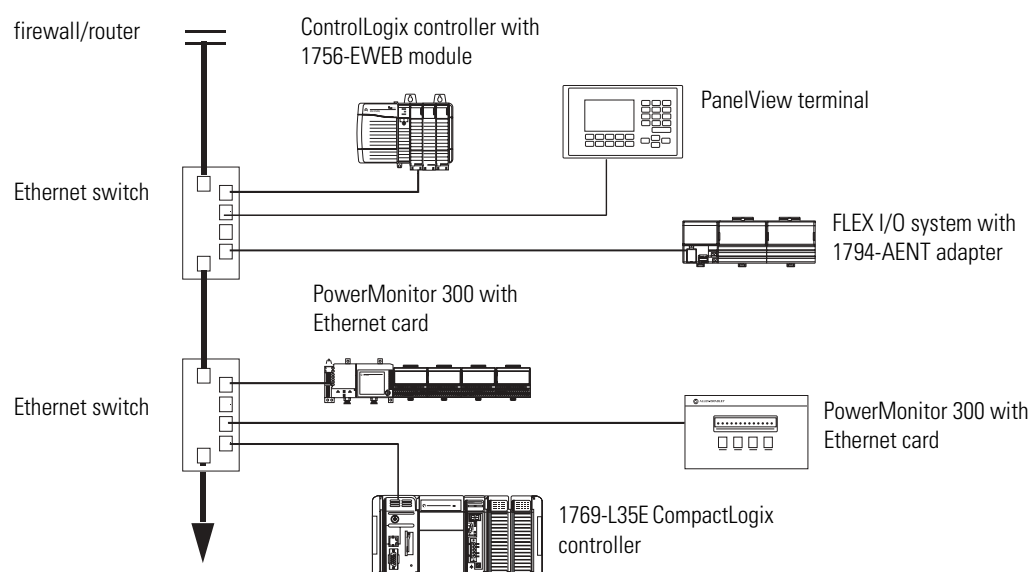
- custom web pages

Create custom web pages that are tailored to your application. Use ASP functions to populate your web pages with live controller data.

- email capability

You can initiate email messages from the embedded email composer in the web server module. You can also use the web server module to send an email initiated by a Logix controller via a MSG instruction.

The following diagram shows how the web server module might fit in your ControlLogix control system on an EtherNet/IP network:



If both ControlLogix chassis in this sample system contain a 1756-EWEB module, the PC could access either 1756-EWEB to monitor and modify data in the ControlLogix controllers.

Browser Requirements

You can access the web server modules with any standard web browser. To access data view pages, the browser requires XML, XSL, and Javascript support. Such browsers include:

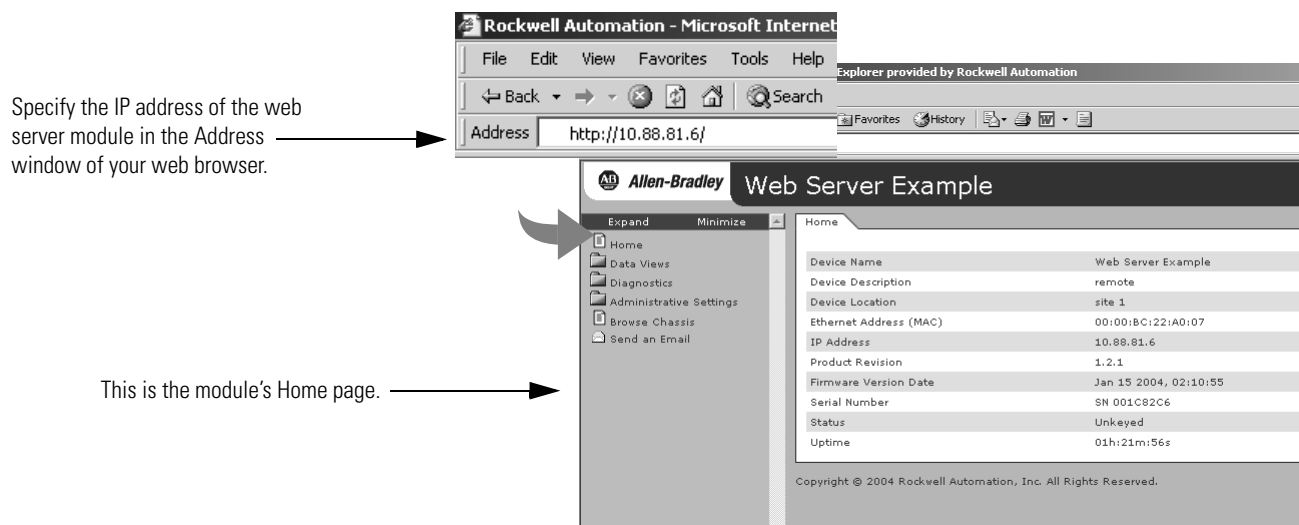
- Internet Explorer 6 and greater
- Netscape 7 and greater
- Mozilla 1.1 and greater

To create and edit data views, you need Internet Explorer 6 or greater.

The supported display size is 640 x 480 or greater. Smaller display sizes work but might require extensive scrolling to view the information.

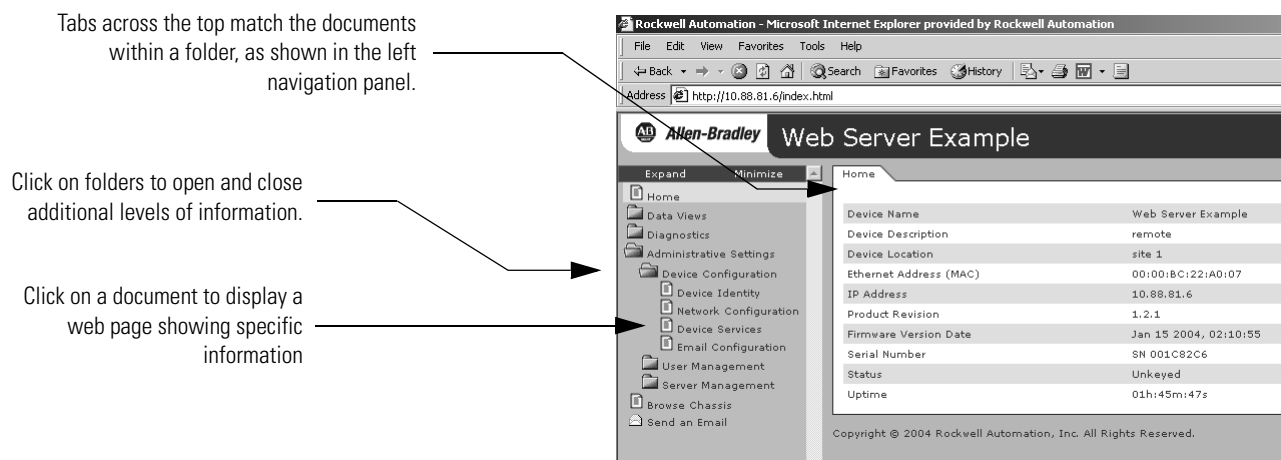
Accessing the Home Page

From your web browser, enter the IP address of the web server module. The module displays its Home page



Navigating the Web Server Module

You navigate the web server's web pages using the navigation panel on the left of the screen. There are also tabs across the top you can use to navigate the sections within folders.



Network Addressing for a Web Server Module

How to Use This Chapter

This chapter describes how to configure a web server module to operate on an Ethernet network.

For this information:	See page:
Determining Which Network Parameters Are Required	2-1
Assigning Network Parameters when the Network HAS a DHCP Server	2-3
Assigning Network Parameters when the Network DOES NOT HAVE a DHCP Server	2-5
DNS Addressing	2-10
Verify Network Settings	2-12

Determining Which Network Parameters Are Required

In order for the web server module to operate on an Ethernet network, you must define these parameters:

Ethernet Parameter:	Description:
IP address	<p>The IP address uniquely identifies the module. The IP address is in the form xxx.xxx.xxx.xxx where each xxx is a number between 0-255. These are reserved values you cannot use:</p> <ul style="list-style-type: none">• 127.0.0.1• 0.0.0.0• 255.255.255.255
subnet mask	<p>Subnet addressing is an extension of the IP address scheme that allows a site to use a single network ID for multiple physical networks. Routing outside of the site continues by dividing the IP address into a net ID and a host ID via the class. Inside a site, the subnet mask is used to redivide the IP address into a custom network ID portion and host ID.</p>
gateway	<p>A gateway connects individual physical networks into a system of networks. When a node needs to communicate with a node on another network, a gateway transfers the data between the two networks.</p>

If you use the web server module to initiates MSG instructions that use host names or to initiate emails, you must also define these parameters:

Ethernet Parameter:	Description:
host name	A host name is part of a text address that identifies the host for a module. The full text address of a module is <i>host_name.domain_name</i> .
domain name	<p>A domain name is part of a text address that identifies the domain in which the module resides. The full text address of a module is <i>host_name.domain_name</i>. The domain name has a 48-character limit.</p> <p>If you specify a DNS server, you must enter a domain name. Also, if you send email from the module, some mail relay servers require a domain name be provided during the initial handshake of the SMTP session.</p>
primary DNS server address	This identifies the DNS server(s), if used in the network. You must have a DNS server configured if you specified a domain name or a host name in the module's configuration. The DNS server converts the domain name or host name to an IP address that can be used by the network.
secondary DNS server address	

Check with your network administrator to determine if you need to specify all of the above parameters.

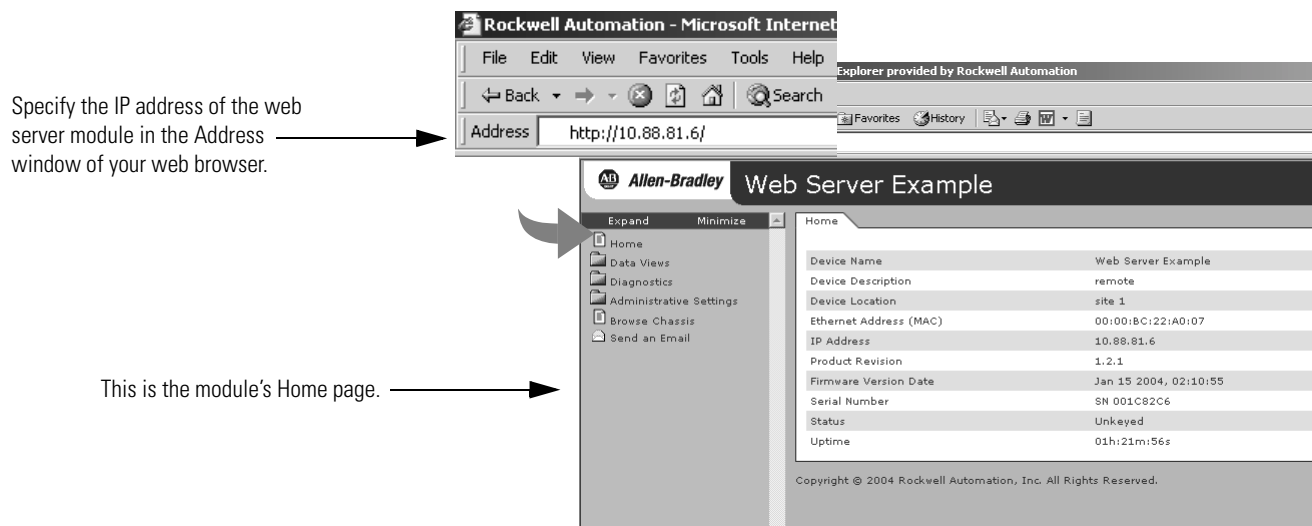
How you configure these network parameters depends on whether or not the Ethernet network has a DHCP server.

Assigning Network Parameters when the Network HAS a DHCP Server

By default, the web server module is DHCP enabled. DHCP (Dynamic Host Configuration Protocol) software automatically assigns IP addresses to client stations logging onto a TCP/IP network.

If you connect the web server module to a network that has a DHCP server, that server will assign an IP address to the web server module and the four-digit display on the front of the web server module will display each of the four numbers of the IP address.

From your web browser, enter this IP address. The module displays its Home page:



The IP address from the DHCP server provides initial access to the web server module. Check with your network administrator whether you need to modify the IP address for future access to the module. The network administrator might have you:

- convert the initial IP address to a static IP address
- enter a different, unique IP address and configure that new address as a static address
- do nothing because the DHCP server was configured so that the initial IP address is already permanently assigned to the web server module

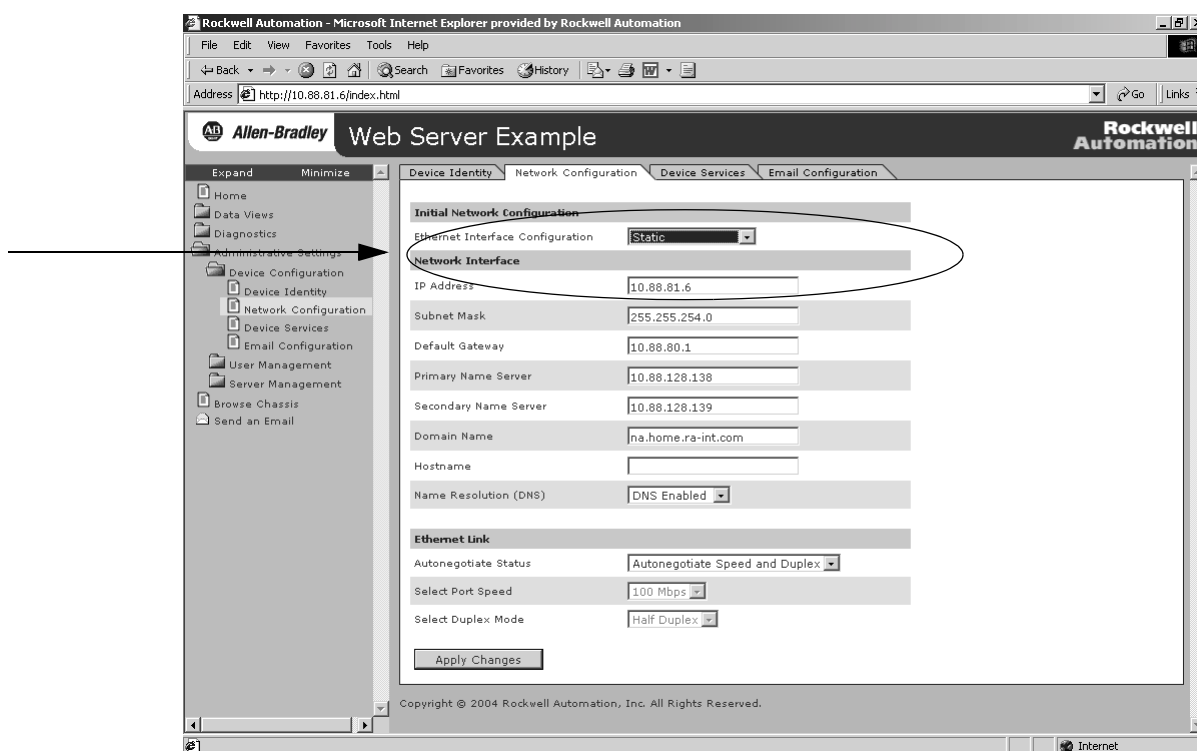
Assigning a static IP address

If your environment requires a static IP address, configure the IP address on the Administrative Settings → Device Configuration → Network Configuration page.

IMPORTANT

Do not simply configure the initial address assigned by the DHCP server as a static IP address. Contact your network administrator for an appropriate static IP address.

On the Network Configuration page:



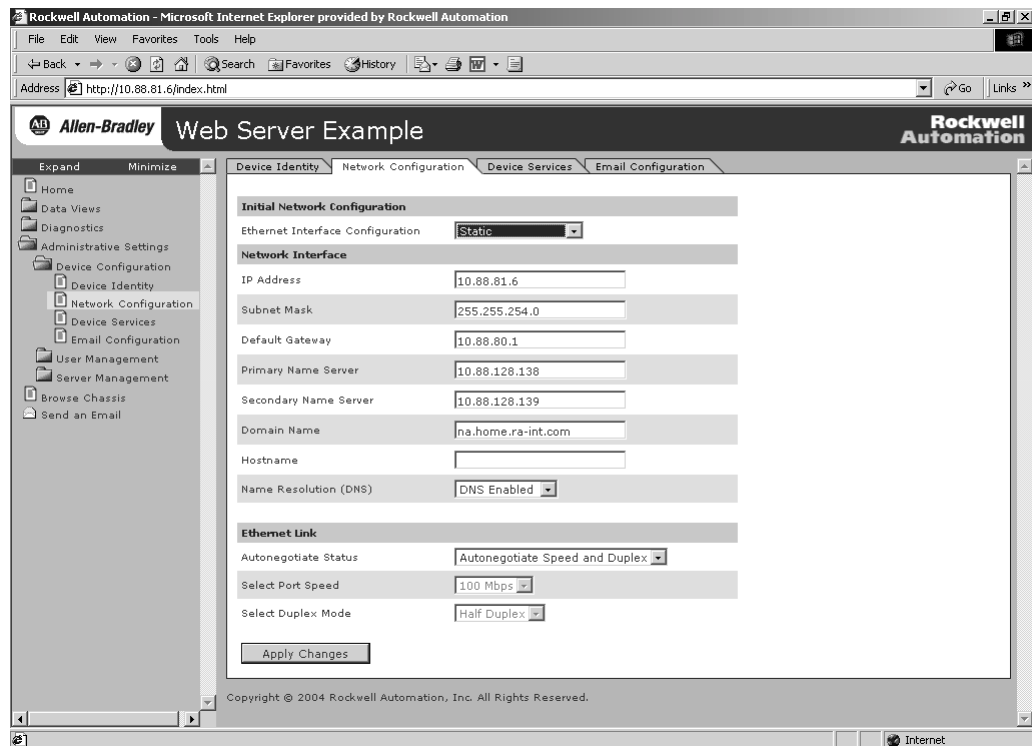
- enter the static IP address
- select static for the Ethernet Interface Configuration

Assigning Network Parameters when the Network DOES NOT HAVE a DHCP Server

If a DHCP server is not available, you must assign a static IP address to the web server module. Select one of these methods:

If you are working in these conditions:	Use this method for assigning network parameters:	See page:
<ul style="list-style-type: none"> in any conditions, the Rockwell Automation BOOTP/DHCP utility is recommended 	Rockwell BOOTP/DHCP utility (available with RSLinx and RSLogix 5000 software)	2-6
<ul style="list-style-type: none"> the module is connected to other NetLinx networks 	RSLinx software	2-8
<ul style="list-style-type: none"> the RSLogix 5000 project is online with the controller that communicates to or through the module 	RSLogix 5000 software	2-9

After using one of these utilities, you can use the Administrative Settings → Device Configuration → Network Configuration page on the web server module to set additional parameters.



Using the Rockwell Automation BOOTP/DHCP utility

The web server module ships with DHCP enabled. The BOOTP/DHCP utility is a stand-alone program that lets you interactively define the IP address of a module that is issuing DHCP or BOOTP requests. The utility is located in the:

- BOOTP-DHCP Server folder in the Rockwell Software program folder on the Start menu (the utility is automatically installed when you install RSLinx software)
- Tools directory on the RSLogix 5000 installation CD.

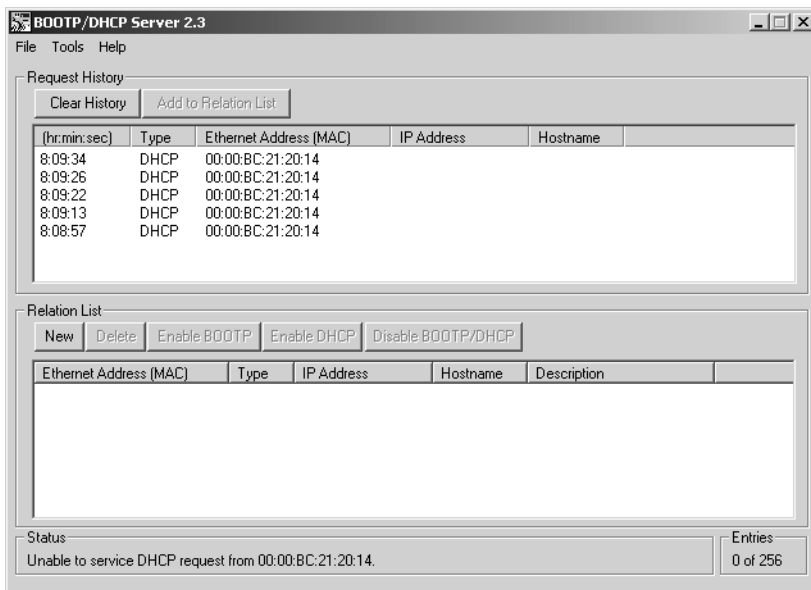
IMPORTANT

Before you start the BOOTP/DHCP utility, make sure you have the hardware (MAC) address of the web server module. The hardware address is on a sticker located on the side of the module. The hardware address is in a format similar to: 00-0b-db-14-55-35.

To use the BOOTP/DHCP utility:

1. Start the BOOTP/DHCP software.

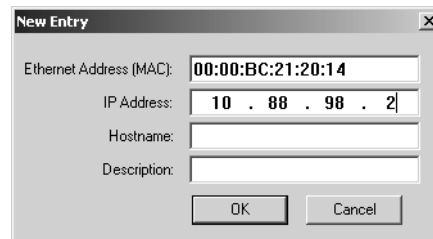
In the Request History panel you see the hardware addresses of modules issuing requests.



2. Double-click on the hardware (MAC) address of the module you want to configure.

The hardware address is on a sticker located on the side of the web server module. The hardware address will be in a format similar to: 00-0b-db-14-55-35.

The New Entry window displays the MAC address you selected and prompts you to enter the IP address.

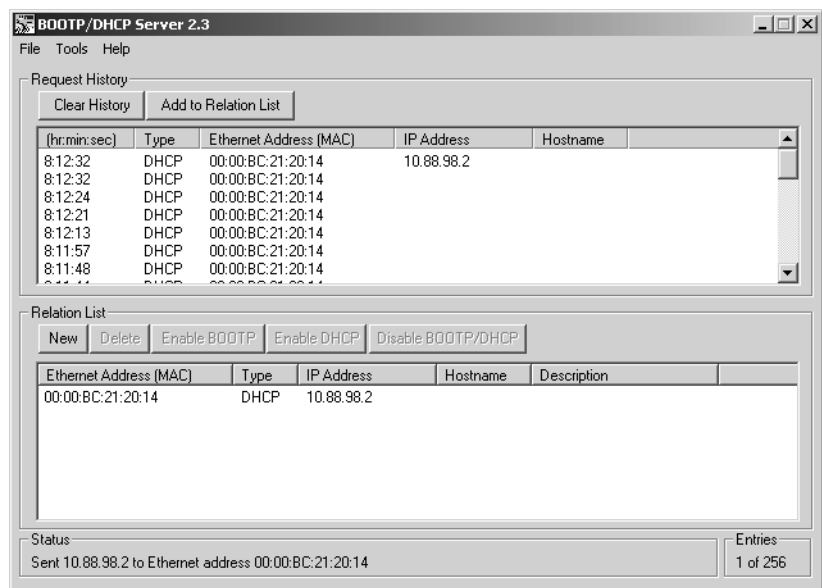


The 'New Entry' dialog box contains the following fields and buttons:

- Ethernet Address (MAC):** 00:00:BC:21:20:14
- IP Address:** 10 . 88 . 98 . 2
- Hostname:** (empty field)
- Description:** (empty field)
- Buttons:** OK, Cancel

3. Enter the IP address of the module. You can also enter the host name and a description of the module. Click OK.

The device is added to the Relation List.



The 'BOOTP/DHCP Server 2.3' window shows the following components:

- Request History:** A table with columns (hr:min:sec), Type, Ethernet Address (MAC), IP Address, and Hostname. It contains several entries for DHCP requests from MAC 00:00:BC:21:20:14 to IP 10.88.98.2.
- Relation List:** A table with columns Ethernet Address (MAC), Type, IP Address, Hostname, and Description. It contains one entry: MAC 00:00:BC:21:20:14, Type DHCP, IP Address 10.88.98.2.
- Status:** Sent 10.88.98.2 to Ethernet address 00:00:BC:21:20:14
- Entries:** 1 of 256

4. To permanently assign this configuration to the web server module, highlight the module and click on the Disable BOOTP/DHCP button. When power is recycled, the module uses the configuration you assigned and does not issue a request.

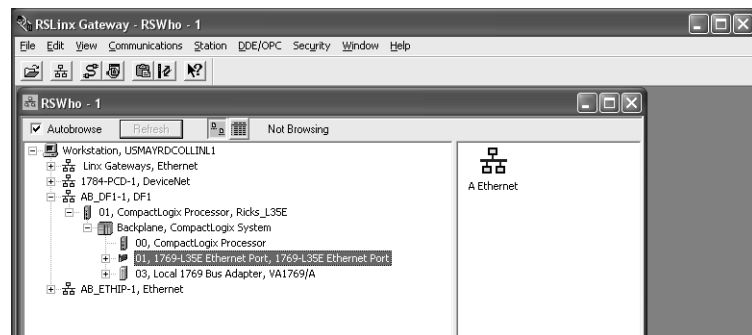
If you do not select the Disable BOOTP/DHCP button, on a power cycle, the web server module clears the current IP configuration and will again begin sending requests.

If you use the BOOTP/DHCP utility in an uplinked subnet where an enterprise DHCP server exists, the module may get an IP address from the enterprise server before the BOOTP/DHCP utility even sees the module. To avoid this, disconnect from the uplink to set the address and have the module remember its static address before reconnecting to the uplink. This is not a problem if you have node names configured in the module and leave DHCP enabled.

Using RSLinx software to set the IP address

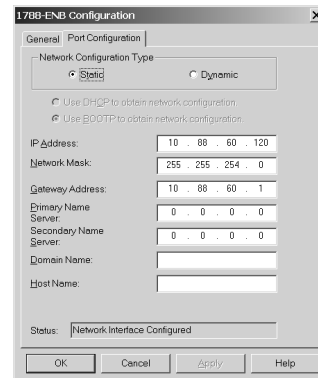
To use RSLinx to configure the IP address.

1. Make sure the web server module is installed and powered up.
2. Start RSLinx. The RSWho window opens. Navigate in RSWho to the web server module.
3. Right-click on the web server module and select Module Configuration.



4. Select the Port Configuration tab, choose Static for the Network Configuration Type, and enter the IP address and the other network parameters, if needed.

Also, select the Static radio button to permanently assign this configuration to the port. If you select Dynamic, on a power cycle, the controller clears the current IP configuration and will again begin sending requests.

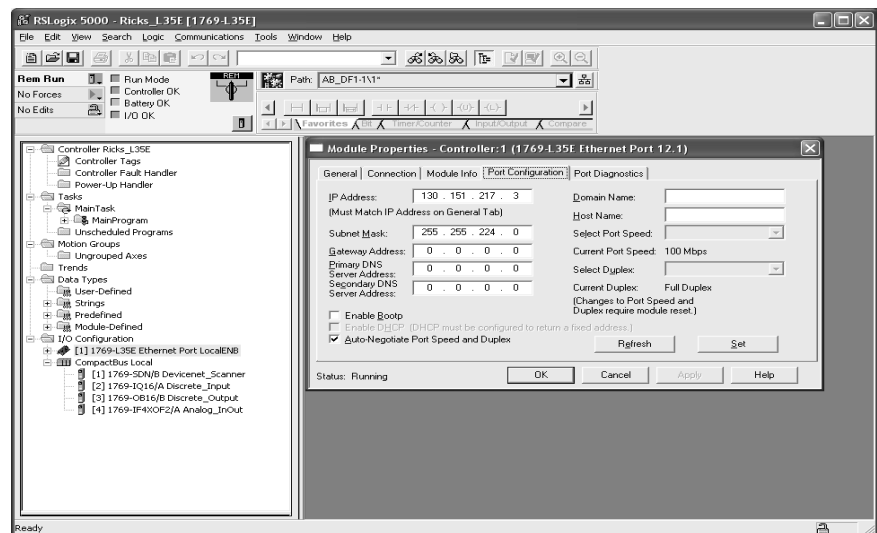


Using RSLogix 5000 software to set the IP address

To use RSLogix 5000 software to configure the IP address:

1. Make sure the module is installed and powered up.
2. Connect to the controller via a serial, or other network, connection.

3. Start RSLogix 5000 software. In the Controller Organizer, select properties for the EtherNet/IP module.



4. Select the Port Configuration tab and specify the IP address and the other network parameters, if needed. Click Apply and then click OK.

This sets the IP address in the hardware. This IP address should be the same IP address you assigned under the General tab.

On this screen, you can also specify port speed (10 Mbps or 100 Mbps) and duplex mode (autonegotiate, half duplex, or full duplex). The module configuration needs to agree with how the switch is configured. See your network administrator for more information.

DNS Addressing

To further qualify an address of a module, you can use DNS addressing to specify a host name for a module, which also includes specifying a domain name and DNS servers. DNS addressing lets you set up similar network structures and IP address sequences under different domains.

DNS addressing is only necessary if you refer to the module by host name and use the web server module to initiate MSG instructions out of the web server module to another device.

To use DNS addressing, you must:

1. Assign a host name to the module.

Your network administrator should be able to assign a host name. Valid host names should be IEC-1131-3 compliant.

2. Configure the module's parameters.

In addition to the IP address, subnet mask, and gateway address, you must also configure a host name for the module, domain name, and primary/secondary DNS server addresses. In the DNS server, the host name must match the IP address of the module.

IMPORTANT

Make sure the DNS enable bit is set

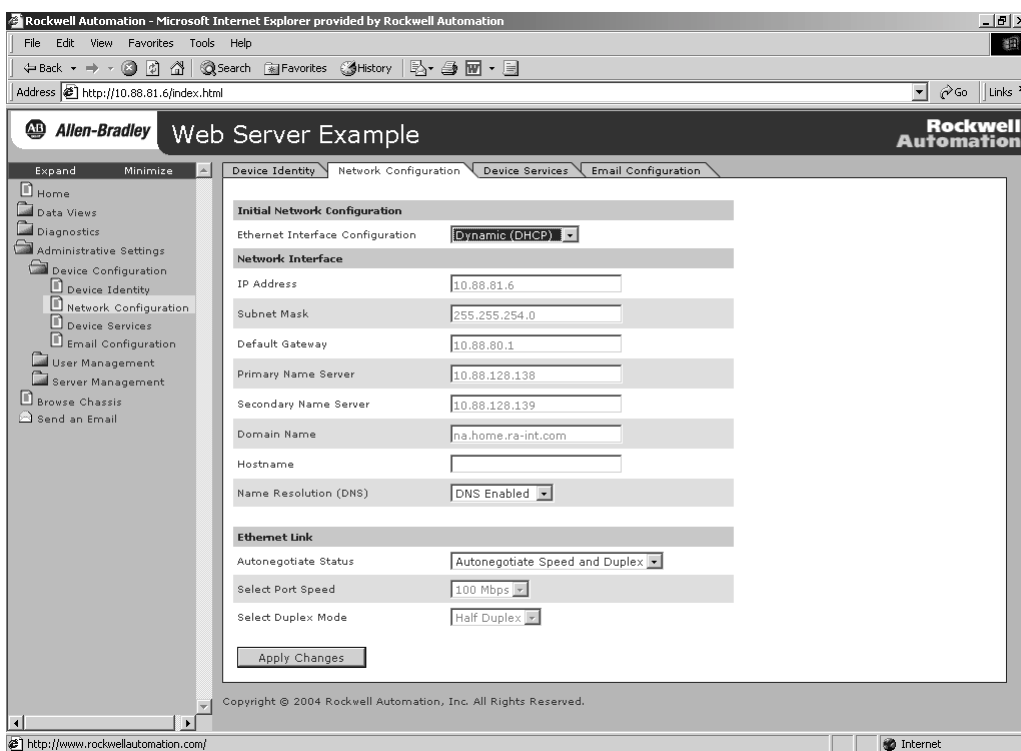
If you configure your module using RSLinx 2.41.00, the enable bit is cleared and DNS addressing will not work. If you configure your module using the Port Configuration tab in RSLogix 5000 software, the enable bit is set, so DNS addressing should work.

3. In RSLogix 5000 software, add the module to the I/O configuration tree and enter the host name in the General tab of the module.

If a child module resides in the same domain as its parent module, just enter the host name. If the child module is in a different domain than its parent module, you must enter the host name and the domain name (host.domain)

Verify Network Settings

The Administrative Settings → Device Configuration → Network Configuration page lets an authenticated user modify network parameters.



In this field:	Specify:
Ethernet Interface Configuration	the network configuration scheme: <ul style="list-style-type: none"> • dynamic BOOTP • dynamic DHCP (default) • static
IP Address	IP address for the web server module If you want to specify a static IP address for the web server module, you must also select Static for the Ethernet Interface Configuration field towards the bottom of this page.
Subnet Mask	subnet mask for the web server module
Default Gateway	gateway address for the web server module
Primary Server Name	DNS server names, if using DNS addressing
Secondary Server Name'	
Domain Name	domain name for the web server module, if using DNS addressing
Host Name	host name for the web server module, if using DNS addressing
Name Resolution (DNS)	whether or not the web server module uses DNS addressing

In this field:	Specify:
Autonegotiate Status	how to determine port speed and duplex: <ul style="list-style-type: none">• autonegotiate speed and duplex• force speed and duplex
Select Port Speed	port speed (10 Mbps or 100 Mbps), if you selected to force speed and duplex
Select Duplex Mode	duplex (full or half), if you selected to force speed and duplex

Notes:

Managing Module Settings

How to Use This Chapter

This chapter describes how to configure module settings other than network parameters for the web server module.

For this information:	See page:
Managing Module Information	3-1
Managing Server Settings	3-5

Managing Module Information

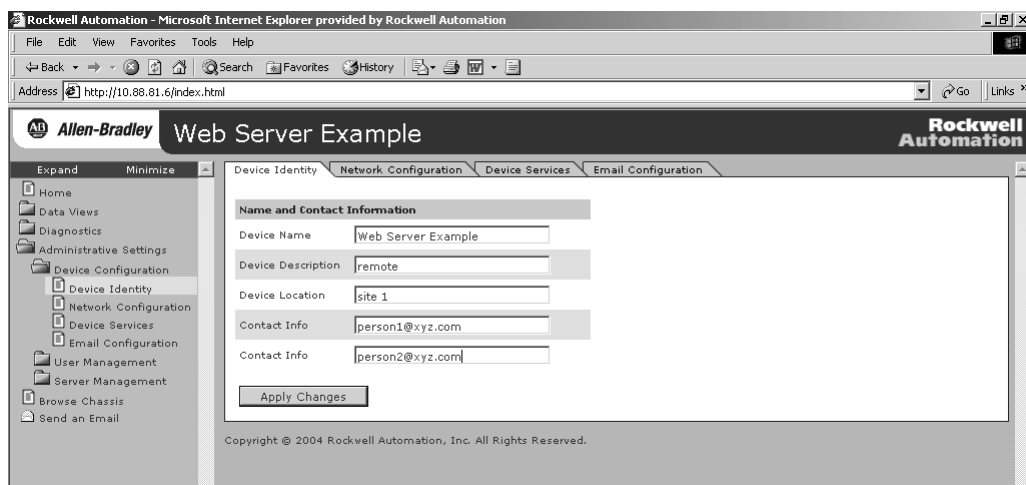
The Administrative Settings → Device Configuration folder provides access to modify module-specific information. You can:

- define the module-specific information that displays on the Home page
- modify network parameters (see chapter 3 for more information about network parameters)
- enable/disable communication services

These module settings are stored in flash memory and persist over power cycles.

Define module-specific information for the Home page

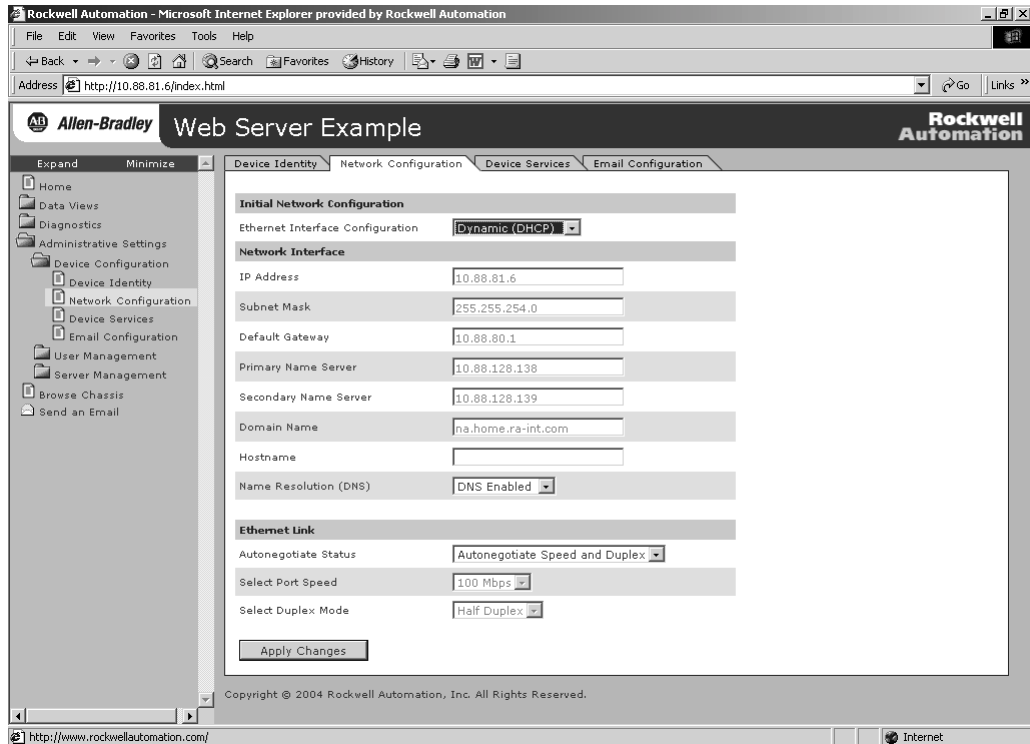
Use the Administrative Settings → Device Configuration → Device Identity page to set specific text that identifies the module. This information appears on the Home page.



In this field:	Specify:
Device Name	<p>name for the web server module 32 characters maximum</p> <p>The device name you enter appears in the title bar of the web server module's web pages. This device name also appears in RSLinx when you browse the network.</p>
Device Location	<p>description of the location of the web server module 64 characters maximum</p>
Device Description	<p>description of the web server module 64 characters maximum</p>
Contact Information	<p>contact information, such as name, phone number, or email address 512 characters maximum</p> <p>There are two fields so that you can specify contact information for two individuals.</p>

Modify network parameters

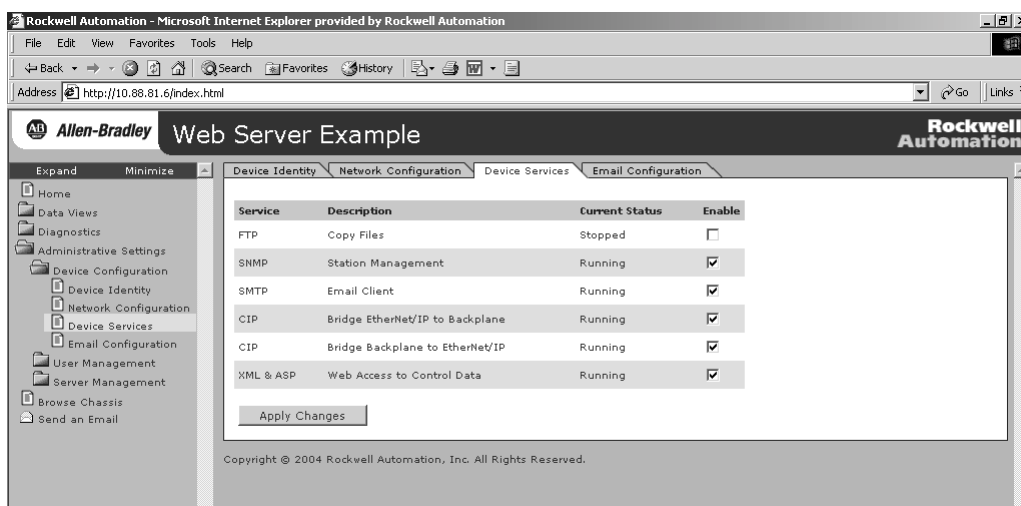
The Administrative Settings → Device Configuration → Network Configuration page lets an authenticated user modify network parameters.



See chapter 2 for more information about network parameters.

Enable and disable communication services

The Administrative Settings → Device Configuration → Device Services page lets you specify which communication services are enabled or disabled on the web server module.



In this field:

Select whether to enable or disable the:

FTP

FTP (File Transfer Protocol) server

Disable FTP to prevent users from accessing the file system on the web server module.

Important: For security purposes, keep FTP disabled unless you frequently transfer files to or from the web server module.

SNMP

SNMP (Simple Network Management Protocol) agent

Enable SNMP if your system uses SNMP management software.

SMTP

SMTP (Simple Mail Transfer Protocol) agent

SMTP manages email capability. Disable SMTP if you do not send emails from the web server module.

CIP Bridge Ethernet to Backplane

CIP (Common Industrial Protocol) bridging

Enable this CIP bridging to allow EtherNet/IP devices to bridge through the web server module to devices in the chassis.

CIP Bridge Backplane to Ethernet

CIP (Common Industrial Protocol) bridging

Enable this CIP bridging to allow other devices in the chassis to bridge through the web server module to EtherNet/IP devices.

XML/ASP

XML/ASP (Extended Markup Language/Active Server Page) support

Enable XML/ASP to allow web access to control system data.

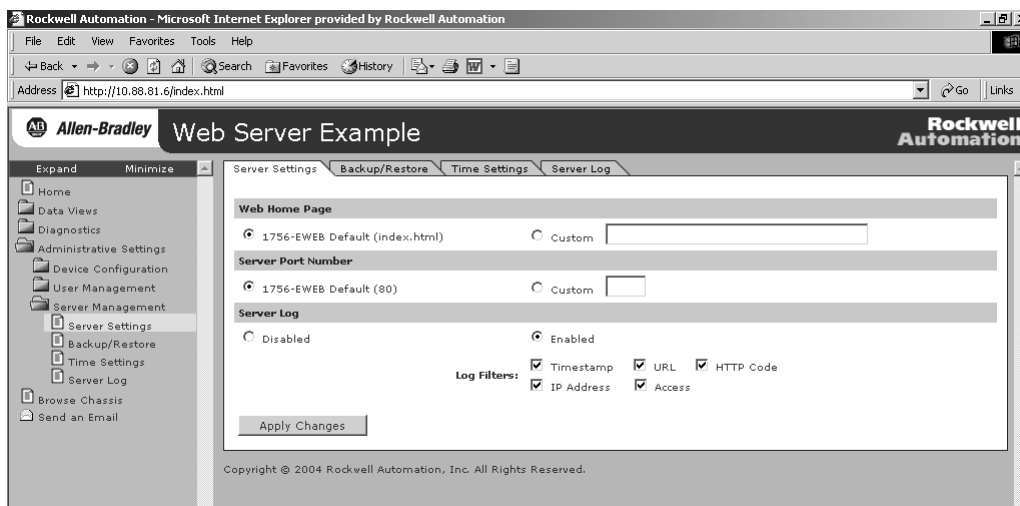
Managing Server Settings

The Administrative Settings → Server Management pages let you customize some of the server settings of the module, as well as back up the file system on the web server module. You can:

- customize server settings, including web home page
- lock access to the module during backup or restore procedures (see chapter 7 for more information on backing up the web server module)
- configure the time server
- display a server log

Customize server settings

The Administrative Settings → Server Management → Server Settings page lets you customize the home web page and server settings of the web server module.



In this field:

Select:

Web Home Page

whether to use the default home page (index.html) or a custom home page

Specify a custom web page as:

/user/web/*mypage*.html where *mypage* is the name of the file for the custom web page.

You must copy a custom home page to the web server module before you can use it. See chapter 8 for information on creating a custom web page.

Server Port Number

whether to use the default port number (80) for the HTTP port on the web server module or to use a custom port number

Server Log

whether to enable or disable the server log

You view the server log from the Server Log page under the Server Management folder. See page 3-8 for how to display the server log.

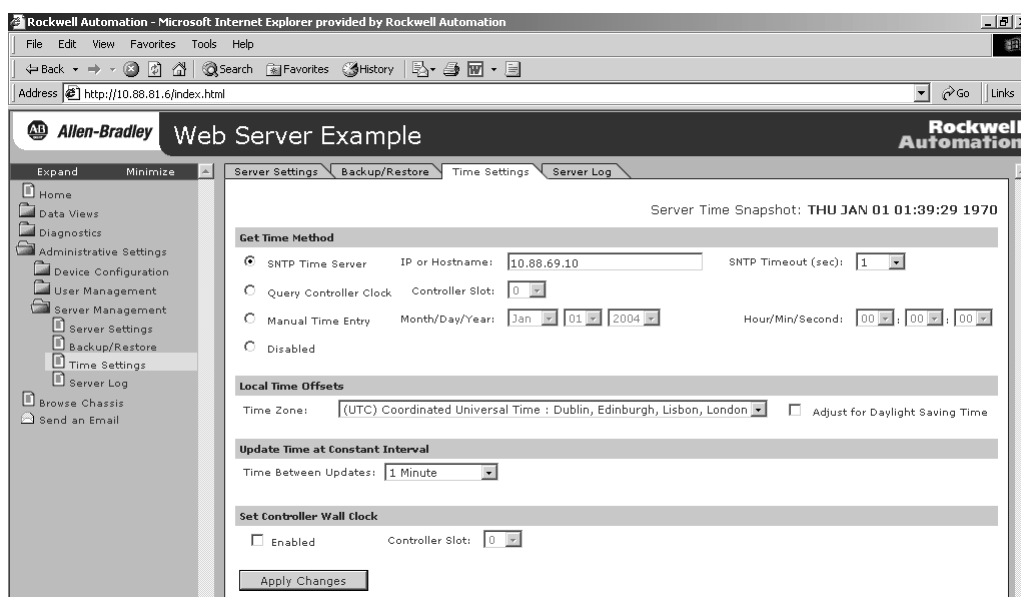
Log Filters

whether you want the following information to be recorded for the web server log:

- timestamp of HTTP request (access)
- URL requested on the web server module
- server HTTP code
- IP address of the requestor
- privilege class access (Administrator, Write, or Read)

Configure the time server

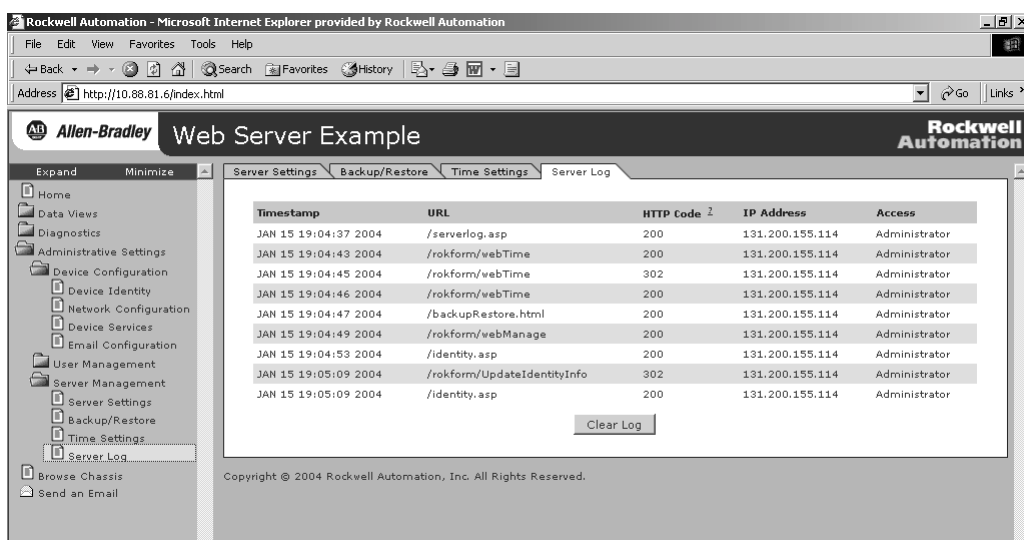
The Administrative Settings → Server Management Server → Time Settings page lets you modify the internal clock in the web server module. This makes sure that files you save to the web server module have accurate date and time stamps. You can also enable the local controller to get its time and date from the web server module.



Select this field:	If you want to:
SNTP Time Server	use the time from the Network Time Protocol (NTP). Specify the IP address or host name of an SNTP server on the network. The web server uses port 123 for this service. The IP address you enter persists over power cycles.
Query Controller Clock	use the time from the local Logix5000 controller (Wall Clock Time) Specify the slot number of the controller. The web server module queries the Wall Clock Time of the controller for both time and date. At subsequent power ups, the web server module queries the controller.
Manual Time Entry	manually set the time and date Manual settings do not persist over power cycles.
Local Time Offsets	select the appropriate time zone This selection is only available when you select SNTP time server as your "Get Time Method."
Update Time at Constant Interval	select how often the web server module updates its date and time
Set Controller Date/Time	use the date and time in the web server module to set the date and time in the local controller You must also specify the slot number of the local controller.

Display the server log

The Administrative Settings → Server Management Server → Server Log page, when enabled, displays records of web accesses to the web server module. Only those information fields that are enabled on the Server Setting page (see page 3-6) appear in the server log. The information displayed on this page is stored in RAM and does not persist over power cycles.



This field:	Specifies:
Timestamp	timestamp of HTTP request (access)
URL	requested URL on the web server module
HTTP code	HTTP code request
IP address	IP address of the requestor
Access	type of access

The web server module has 30K memory allocated for server log entries. If all the log options are enabled, the server log memory can hold about 200 entries. Once this allocation is full, the web server module stops storing server log entries. Press the Clear Log button to empty the server log so that the web server module can again log entries.

Using Data Views to Access Controller Data

How to Use This Chapter

The web server modules provide access to controller data. You can monitor and modify data in controller tags. This chapter shows you how to set up data views of controllers tags.

For this information:	See page:
Overview of Data Views	4-1
Creating a Data View	4-3
Monitoring Data Views and Tag Data	4-5
Creating Data Views Offline	4-8
Editing a Data View	4-7
Using an External Application to Access Data Views	4-9
Example: Data View XML	4-10
Example: Data View XML With Tag Values	4-11
Example: Data View XML With Tag Errors	4-12

For data views, the web server module must be in the same chassis as the controller.

Overview of Data Views

Data views give you the ability to read from and write to Logix5000 controller tags from a browser interface or an external application. The web server module provides web pages that let you configure a set of tags (a “data view”) that can be read and/or written.

A data view consists of an XML file with data tag information. The XML file is in a readable ASCII format. It contains the tag name, data type, path, display formatting, and privilege access level. Each tag value is exposed as a separate element and an error attribute is optional.

Tags supported in data views

To configure tags in data views:

- You can only access tags in controllers that reside in the local chassis (same chassis as the web server module).
- Tags must be controller-scoped.
- Tags must be an atomic type (BOOL, SINT, INT, DINT, REAL, STRING). You can specify a member of a structure or an array, but you cannot specify an entire structure or array.
- A tag can only appear once in a particular data view. You cannot, for example, have two instances of the same tag with different display formats.
- There is no limit to the number of data views as long as the total number of entries in all data views on one web server module does not exceed 2500 entries. Each tag you configure on a data view is one entry. If you configure the same tag in multiple data views, each tag is considered one entry.

Performance estimates

For access to the XML data views, the web server module can produce data according to the table below. This table assumes the absence of significant CIP traffic and does not take into account the amount of the time for the browser to render the data view page.

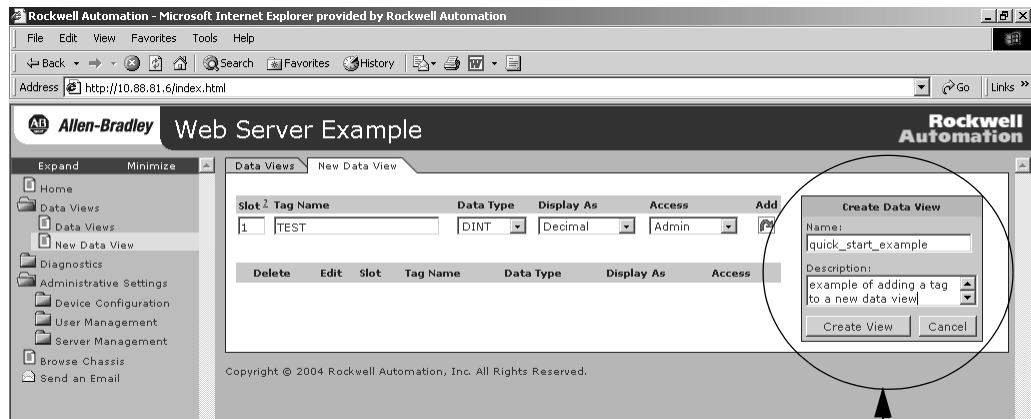
Tags per Data View:	Time per Data View:
10	100 ms
100	350 ms
1000	3 sec

Creating a Data View

Each data view contains a group of tags that you want to monitor. Each web server module can support multiple data views.

You create a data view from the Data Views → New Data View page.

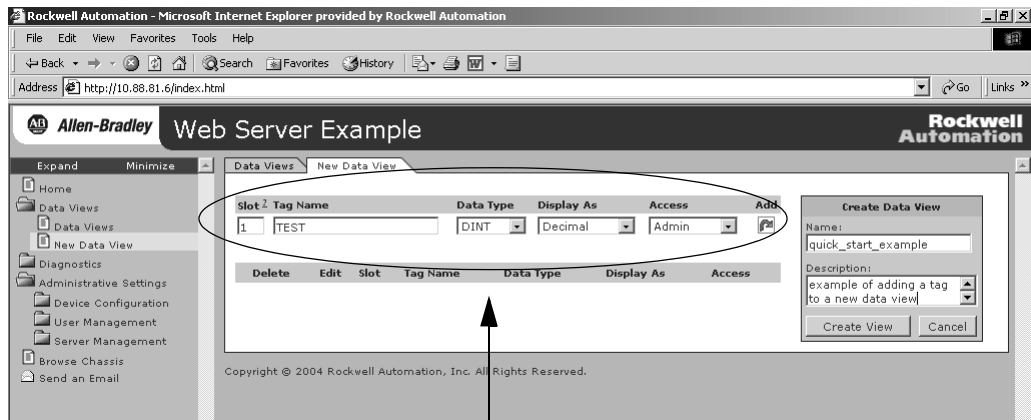
1. Use the Create Data View window on the right of the window to identify the data view.




specify a data view here

Enter a name (required) and a description (optional) for the data view.

2. Add at least one tag to the data view.



specify a tag here

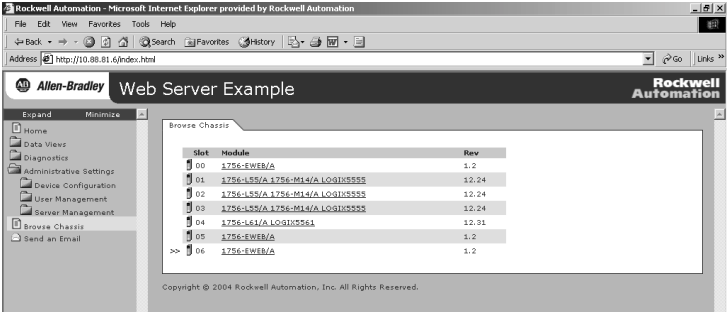
3. Click the Add button  to add the tag you just specified.

You can add multiple tags to the data view, as long as there are no more than 2500 tags in all the data views in one web server module.

4. Click the Create View button to create the data view.

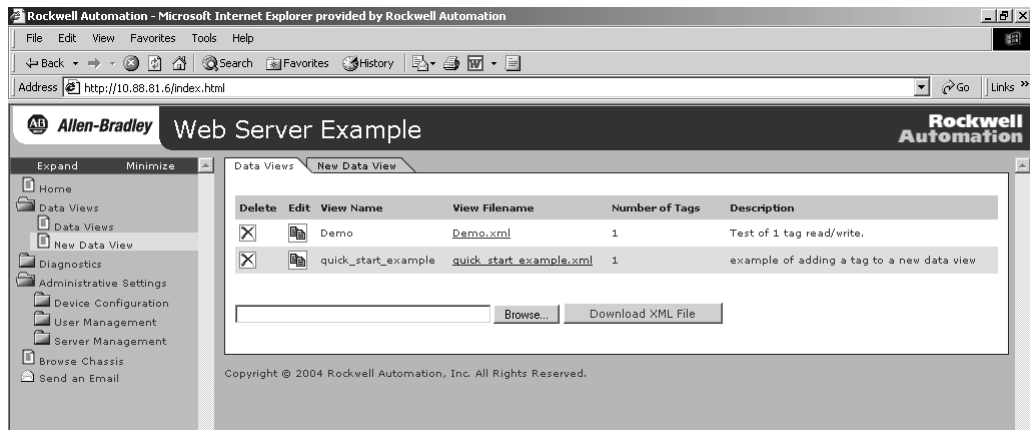
Adding tags to a data view

When you add a tag to a data view, you specify:

In this field:	Specify:	Details:
Slot	slot number of the controller	Click on the question mark next to Slot slot ? or use the Chassis Browse page to validate the controller slot. 
Tag Name	name of the tag	These fields must match exactly what is specified for the tag in the selected controller. To verify tag information, you can use: <ul style="list-style-type: none">• RSLogix 5000 software to view the controller project• RSLinx software to navigate to the controller and view tags
Data Type	data type of the tag	
Display As	display type to use for the tag	
Access	whether you require Administrator, Write, or Read access to view the tags in this data view	The default access level is Administrator. The access you specify applies to the whole data view, not just the tag. If you have multiple tags with different access levels in the same data view, the web server assigns the highest (most access) level to the data view. See chapter 6 for details on access levels.

Monitoring Data Views and Tag Data

Use the Data Views → Data Views page to view existing data views. Click on the file name to view the tags within a data view.



The data view displays in an XML format using an XSL style sheet. To quickly access the XML file, right-click in the data view and in:

- Internet Explorer, select View Source and save the resulting text
- Netscape or Mozilla, select This Frame → Save As

You can also use the backup/restore function to FTP a copy of the XML file. See chapter 7.

If the fields specified for the tag don't match the tag as it is specified in the controller, this page indicates an error and the tag value shows xsi:nil for its value.

From this page, you can modify the value of a tag if you have Administrator or Write access. Enter the new value and click Update. There is no auto-refresh for the data view pages so as to not impact controller execution.

Sorting data views

You can sort data views by:

- name (alphabetically)
- filename (alphabetically)
- number of tags (numerically)
- description (alphabetically)

Click the column title (such as View in the blue heading of the table of data views).

Click a column name to sort the column.



The first click sorts in ascending order. Click again to sort in descending order. An arrow next to the column name shows the direction of the current sort.


You can also sort the tags within a data view by clicking on the slot, tag name, data type, display as, value, or access headings in the column title.

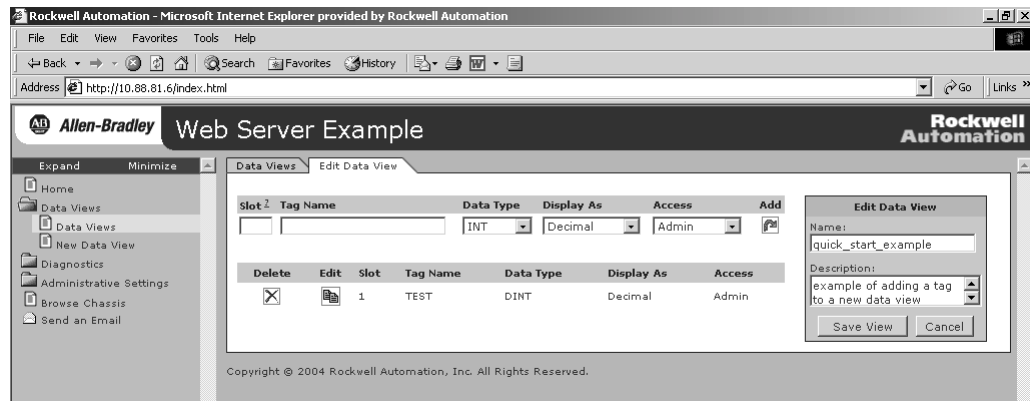
Interfacing with the Logix5000 controller


When you request to display a data view, the web server module establishes one connection to the target controller. Tag values are read and written over this connection. After the web server module retrieves the data view or updated the data view, the web server module closes the connection.

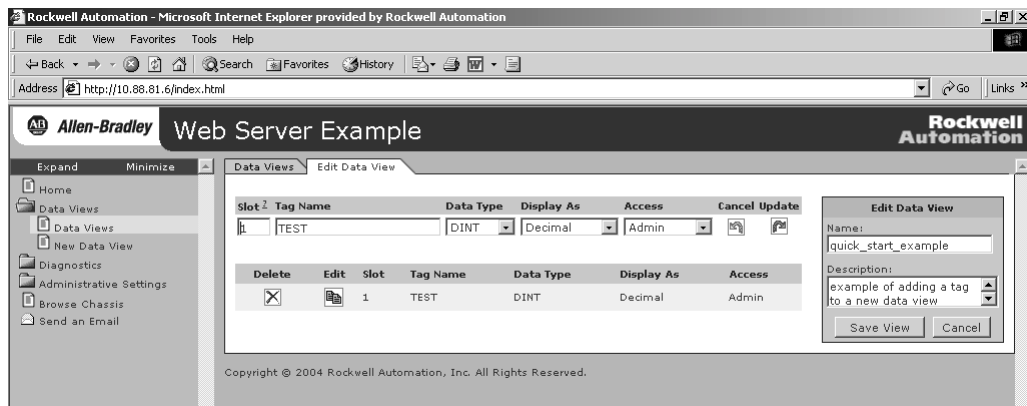
If someone changes tag names in the target controller and those tags exist in a data view, the data view will display an error message indicating that the tag was not available.

Editing a Data View

To edit an existing data view, click the edit symbol  next to the data view you want to edit.



From here, you can add additional tags or edit existing tags. To edit an existing tag, click the edit symbol  next to the tag name. When you edit a tag, highlights the tag you selected and populates the tag fields with the already configured information.



Creating Data Views Offline

You can create data views offline as XML files and later copy them into the web server module. To create a data view offline:

1. Use a text editor to create an XML data view file.

Right-click in the data view and in:

- Internet Explorer, select View Source and save the resulting text
- Netscape or Mozilla, select This Frame → Save As

You can also use the backup/restore function to FTP a copy of the XML file. See chapter 7.

2. Scroll to the bottom of the Data Views page on the web server module.
 - a. Use the Browse button to locate the XML data view file.
 - b. Use the Download XML File button to copy the XML data view file to the web server module



Data views are stored in the “/user/system/dataviews/” directory on the web server module.

Using an External Application to Access Data Views

The XML format of data views makes the data views files accessible by user-written programs. Many programming languages, such as Java and Visual Basic, have facilities for processing XML files.

User programs access data views by making HTTP requests. This is just like a web browser, only instead of displaying the data view, the user program processes the XML data. The browser uses an XSL stylesheet to display the XML files. The XSD schema files validate data views.

File Format:	Description:
XSL	<p>An XML data view specifies an external XSL stylesheet that contains the rules for transforming this XML information into HTML. A web browser uses the XSL stylesheet to display the data view.</p> <p>The XSL file is stored in <i>address/dataview/dataview.xml</i> where <i>address</i> is the IP address or host name of the web server module.</p>
XSD	<p>The web server module provides an XML schema (<i>dataview.xsd</i>) for validating data views. This schema also references the <i>CIPDataTypes.xsd</i> schema.</p> <p>The XSD files are stored in <i>address/schema/dataview.xsd</i> and <i>address/schema/CIPDataTypes.xsd</i> where <i>address</i> is the IP address or host name of the web server module.</p>

Read a data view via an external application

For an external application to read a data view, the application issues an HTTP GET command, that specifies the location and filename of the data view.

Data views are located in the */user/system/dataviews* directory. The URL for a data view named “myview” would be:

“http://IP_address/user/system/dataviews/myview.xml”

Change data in a data view via an external application

When an external application completes modifying tag data in a data view, it should post the modified data view, either as a file attachment (in a multi-part form) or in a single form field named “xml”, to the URL of the data view itself. If all the modified tags are successfully written, the web server module redirects the application to the newly modified data view. If any tag cannot be written to the controller, the web server module returns an HTTP error code with a status message indicating the error.

Example: Data View XML

This is an example XML markup for a data view named “alltypes”. The data view contains one tag for each of the supported data types. The tags are in the controller residing in slot 1.

```
<?xml version="1.0"?>
<?xml-stylesheet href="/dataview/dataview.xsl" type="text/xsl"?>
<view
xmlns="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.rockwellautomation.com/technologies/data_access/data_
views/1.0/ /schema/dataview.xsd"
xmlns:cip="http://www.rockwellautomation.com/technologies/data_access/data_types/1.0
/" name="alltypes" description="">
  <tag name="test_tag_bool" valueType="cip:dt_BOOL" path="1,1" display="String"
access="write">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_sint" valueType="cip:dt_SINT" path="1,1" display="Decimal"
access="write">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_int" valueType="cip:dt_INT" path="1,1" display="Decimal"
access="write">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_dint" valueType="cip:dt_DINT" path="1,1" display="Decimal"
access="write">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_real" valueType="cip:dt_REAL" path="1,1" display="Decimal"
access="write">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_string" valueType="cip:dt_STRINGI" path="1,1"
display="String" access="write">
    <value xsi:nil="true"/>
  </tag>
</view>
```

Example: Data View XML With Tag Values

This is an example XML markup for a data view named “alltypes” loaded with current tag values. The data view contains one tag for each of the supported data types. The tags are in the controller residing in slot 1.

```
<?xml version="1.0"?>
<?xml-stylesheet href="/dataview/dataview.xsl" type="text/xsl"?>
<view
xmlns="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/ /schema/dataview.xsd"
xmlns:cip="http://www.rockwellautomation.com/technologies/data_access/data_types/1.0/" name="alltypes" description="">
  <tag name="test_tag_bool" valueType="cip:dt_BOOL" path="1,1" display="String"
access="write">
    <value xsi:type="cip:dt_BOOL">TRUE</value>
  </tag>
  <tag name="test_tag_sint" valueType="cip:dt_SINT" path="1,1" display="Decimal"
access="write">
    <value xsi:type="cip:dt_SINT">123</value>
  </tag>
  <tag name="test_tag_int" valueType="cip:dt_INT" path="1,1" display="Decimal"
access="write">
    <value xsi:type="cip:dt_INT">28416</value>
  </tag>
  <tag name="test_tag_dint" valueType="cip:dt_DINT" path="1,1" display="Decimal"
access="write">
    <value xsi:type="cip:dt_DINT">1459879936</value>
  </tag>
  <tag name="test_tag_real" valueType="cip:dt_REAL" path="1,1" display="Decimal"
access="write">
    <value
xsi:type="cip:dt_REAL">-2478827762357103800000000000000000000000.000000</value>
  </tag>
  <tag name="test_tag_string" valueType="cip:dt_STRINGI" path="1,1"
display="String" access="write">
    <value xsi:type="cip:dt_STRINGI">aazz</value>
  </tag>
</view>
```

Example: Data View XML With Tag Errors

This example a data view named “alltypes” with error messages for tags that could not be retrieved.

```
<?xml version="1.0"?>
<?xml-stylesheet href="/dataview/dataview.xsl" type="text/xsl"?>
<view
xmlns="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/ /schema/dataview.xsd"
xmlns:cip="http://www.rockwellautomation.com/technologies/data_access/data_types/1.0/"
name="alltypes" description="">
  <tag name="test_tag_bool" valueType="cip:dt_BOOL" path="1,1" display="String"
access="write" error="Couldn't read tag!">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_sint" valueType="cip:dt_SINT" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_int" valueType="cip:dt_INT" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_dint" valueType="cip:dt_DINT" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_real" valueType="cip:dt_REAL" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
    <value xsi:nil="true"/>
  </tag>
  <tag name="test_tag_string" valueType="cip:dt_STRINGI" path="1,1"
display="String" access="write" error="Couldn't read tag!">
    <value xsi:nil="true"/>
  </tag>
</view>
```

Sending Email

Using This Chapter

This chapter describes how to send an email.

For this information:	See page:
Overview	5-1
Configuring the Web Server to Send Email	5-3
Sending an Email via the Web Page	5-4
Sending an Email via a Controller-Initiated Message Instruction	5-4

For email, the web server module can be remote or local to the controller.

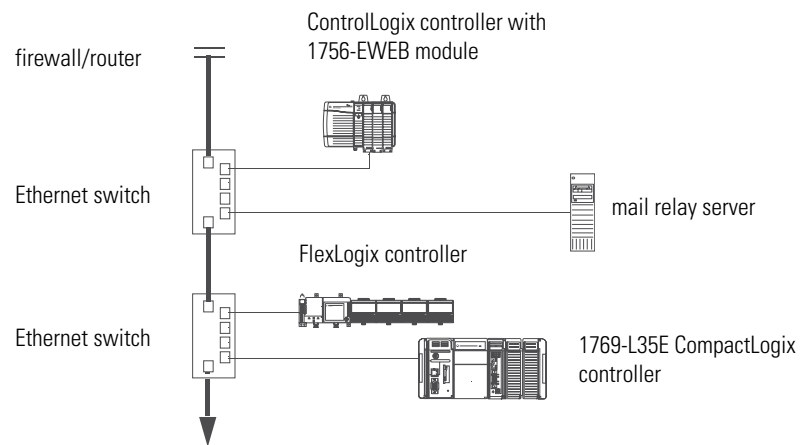
Overview

The web server module is an email client that uses a mail relay server to send email. There are two ways you can use the web server modules to send email.

If you want to:	Then:
send an email to specific personnel when a controller application generates an alarm or reaches a certain condition	program the controller to send a MSG instruction to the web server module
send controller or application status information on a regular basis to a project manager	The MSG instruction then instructs the web server module to send the email text (contained within the MSG instruction) to the mail relay server. Multiple controllers can use the same web server module to initiate email.
test the email configuration of the web server module	use the Send an Email link on the web server's Home page
use the web server email interface to send an email (you must enter all email information each time you use this interface)	

The web server module only sends the content of a MSG instruction (or the content of the message entered on the email web page) as an email to a mail relay server. Delivery of the email depends on the mail server. The web server module does not receive email.

For example, in this sample system:



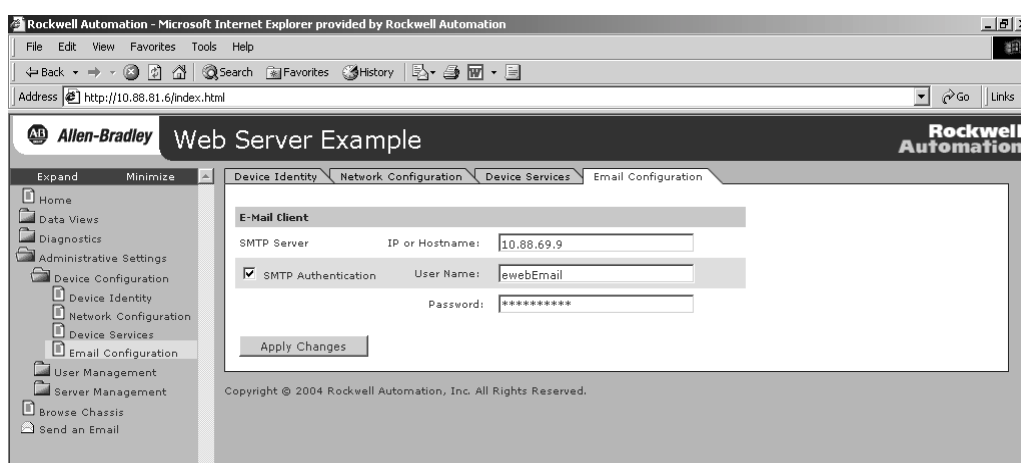
This device:	Can:
ControlLogix controller	send a MSG instruction to the 1756-EWEB web server module to initiate sending an email to the mail relay server
FlexLogix controller	
CompactLogix controller	
1756-EWEB web server module	Use the path of the MSG instruction to identify the web server module as the target of the MSG instruction.
mail relay server	send an email to the mail relay server from the email interface on the Send an Email link.
	Each time you use this interface, you must enter all email information.
	send email to specified recipients
	The mail relay server determines the delivery of any email send through a web server module, whether via a MSG instruction or from its built-in interface.

Configuring the Web Server to Send Email

The web server module uses the standard SMTP protocol to forward an email to a mail relay server. You must configure the web server module to recognize the appropriate mail relay server

Some mail servers require a domain name be provided during the initial handshake of the SMTP session. For these mail servers, make sure you specify a domain name when you configure the network settings for the module.

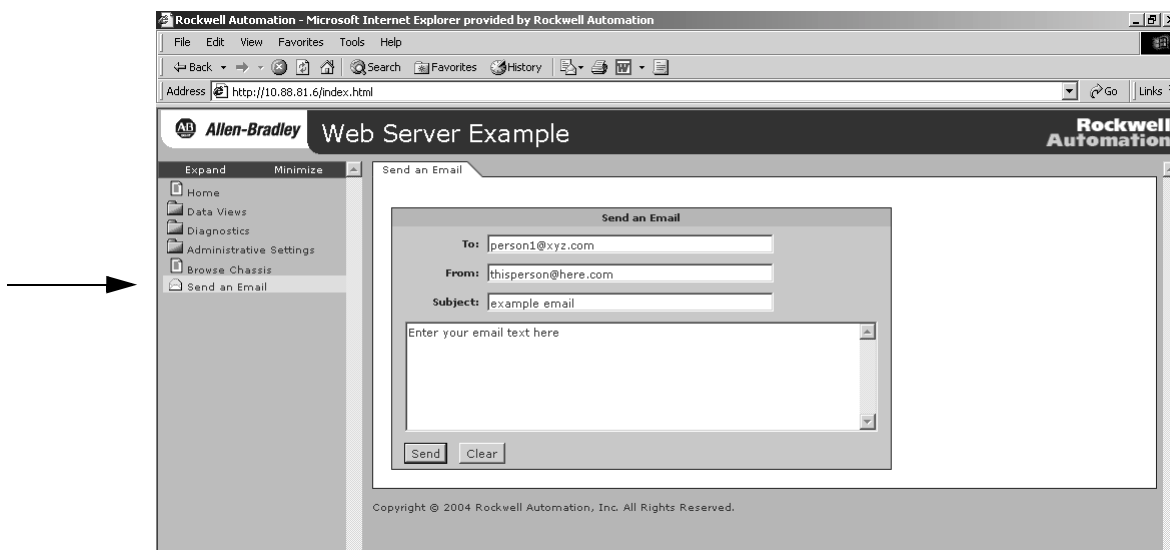
You configure the SMTP server and domain name on the Administrative Settings → Device Configuration → Email Configuration page.



Enter the address of the SMTP server that manages email. You can also select whether or not the web server module should authenticate to the SMTP server. The web server module only supports LOGIN authentication. Check with your network administrator for more information.

Sending an Email via the Web Page

Use the Send an Email link to enter and send email text. This method is one-time approach to sending an email because you have to enter all the email information each time you use this link. This link is most useful for testing the email configuration you specified on the Administrative Settings → Device Configuration → Email Configuration page.



In this field:	Enter the:
To	email address of the recipient
From	email address of the sender
	This address is where you want any replies to this email to go. It is not an email address of the web server module. The web server module only sends email and does not receive email.
Subject	subject line of the email
text window	the email text

Click Send after you specify the email address and enter the text.

Sending an Email via a Controller-Initiated Message Instruction

A Logix controller can send a generic CIP message instruction to the web server module that instructs the web server module to send an email message to a SMTP mail server using the standard SMTP protocol. This is useful to automatically communicate controller data and/or application conditions to appropriate personnel.

IMPORTANT

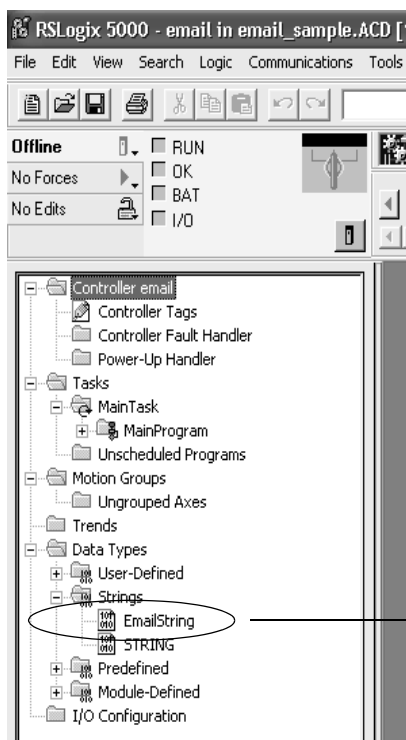
Be careful to write the ladder logic to ensure the MSG instructions are not continuously triggered to send email messages.

Step 1: Create string tags

You need two controller-scoped, string tags:

- one to contain the email text
- one to contain the status of the email transmission

These tags can contain as many as 474 characters. You must create a user-defined STRING data type (the default STRING data type in RSLogix 5000 software is not large enough for most email text). For example, create a STRING data type named EmailString.



Name: EmailString

Description:

Maximum Characters: 520

Members:

Data Type Size: 524

Name	Data Type	Style	Description
LEN	DINT	Decimal	
DATA	SINT[520]	ASCII	

Create one controller-scoped tag of this new data type to contain the email text. Create a controller-scoped second tag of this new data type to contain the transmission status. For example, create tag EWEB_EMAIL (to contain the email text) and EmailDstStr (to contain the transmission status). Both of these tags are of type EmailString.

The screenshot shows the Tag Manager interface with the scope set to 'email(controller)'. The table lists several tags:

Tag Name	Value	Force Mask	Style	Type
+ EmailConfigstring	'10.88.128.111'	{...}		STRING
+ EmailDstStr	'1'	{...}		EmailString
+ EWEB_EMAIL	'To:person1@xyz.com\$r\$1 From: ...'	{...}		EmailString
+ SendEmail_E...	{...}	{...}		MESSAGE
+ SetupMailServer	{...}	{...}		MESSAGE

Annotations on the left:

- tag for status → points to EmailDstStr
- tag for email text → points to EWEB_EMAIL

Annotation below the table:

Click in the Value box to display this button. Click this button to display the String Browser so you can enter the email text.

The String Browser window for EWEB_EMAIL shows the email text:

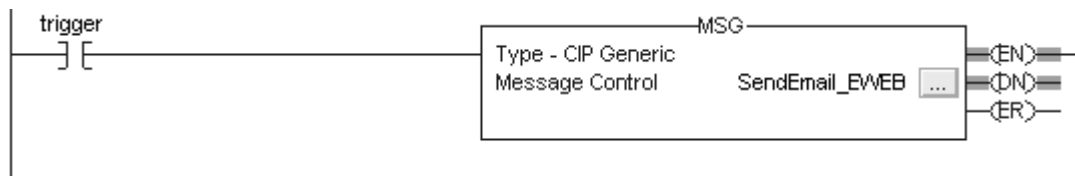
```
To:person1@xyz.com$r$1 From:
1756EWEB@demo.COM$r$1 This is a test!!
$r$1
```

Position: 65 Count: 65 of 520

The text of the email does not have to be static. You can program a controller project to collect specific data to be sent in an email. For more information on using ladder logic to manipulate string data, see the *Logix5000 Controllers Common Procedures Programming Manual*, publication 1756-PM001.

Step 2: Enter the ladder logic

Enter the MSG instruction that triggers the email. Execute this email MSG instruction as often as needed.



Step 3: Configure the MSG instruction

Configure the MSG instruction that contains the email text.

On the Configuration tab of the MSG instruction, configure the MSG parameters for sending an email.

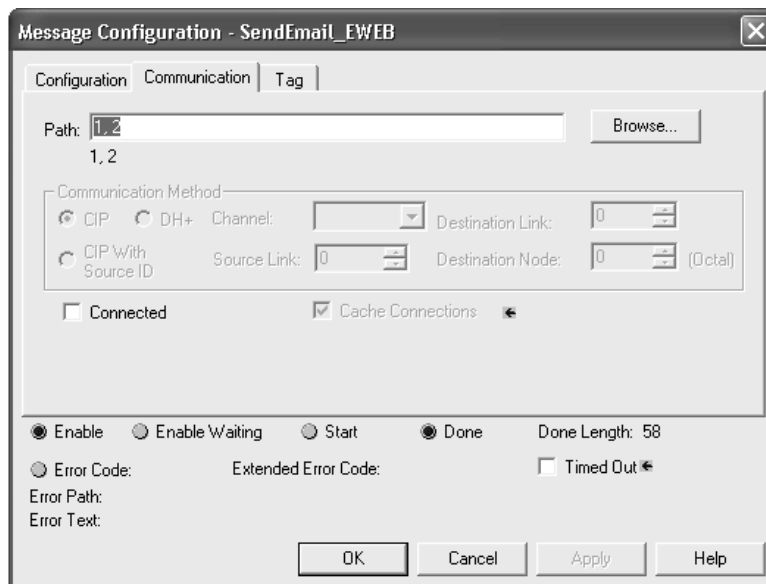
The Source Length is the number of characters in the email tag plus 4 characters.

In this example, the email text contains 65 characters.

where:

In this field:	Enter:
Service Type	Custom
Service Code	4b
Instance	1
Class	32f
Attribute	0
Source Element	the tag that contains the email text
	This tag is of the STRING data type you created to contain the email text. In this example, enter EWEB_EMAIL which is of type EmailString
Source Length	the number of characters in the email text plus 4
	In this example, enter 69 (65 characters in the email + 4)
Destination	a tag to contain the status of the email transmission
	This tag is also of the STRING data type you created to contain the email text. In this example, enter EmailDstStr which is of type EmailString

On the Communication tab of the MSG instruction, configure the path from the controller to the web server module.



The path starts with the controller initiating the MSG instruction module. Then enter the port the message exits and the address of the next module in the path. For example, if the web server module is in the same chassis as the controller and is in slot 2, the path is: 1, 2.

If all the devices in the path are configured in the initiating controller's I/O Configuration tree, you can use the Browse button to select the target web server module and the software automatically fills in the path.

For more information on configuring the path of a MSG instruction, see the *Logix5000 Controllers General Instructions Reference Manual*, publication 1756-RM003.

Entering the text of the email

Use the string browser to enter the text of the email. In the example above, you enter the email text into the EWEB_EMAIL tag. To include “To:”, “From:”, and “Subject:” fields in the email, use <CR><LF> symbols to separate each of these fields. The “To:” and “From:” fields are required; the “Subject:” field is optional. Use a second set of <CR><LF> symbols after the last one of these fields you enter. For example:

```
To: email address of recipient $r$l  
From: email address of sender $r$l  
Subject: subject of message $r$l$r$l  
body of email message
```

The maximum length of an email message is 474 characters. An additional 4-byte string-length value is added to the tag. As a result, the maximum source length is 478 characters.

Possible email status codes

Examine the destination element of the email MSG to see whether the email was successfully delivered to the mail relay server. This indicates that the mail relay server placed the email message in a queue for delivery. It does not mean the intended recipient successfully received the email message. Possible codes that could be in this destination element are:

Error Code (hex):	Extended-Error Code (hex):	Description:
0x00	none	Delivery successful to the mail relay server.
0x02	none	Resource unavailable. The email object was unable to obtain memory resources to initiate the SMTP session.
0x08	none	Unsupported Service Request. Make sure the service code is 0x4B and the Class is 0x32F.
0x11	none	Reply data too large. The Destination string must reserve space for the SMTP server reply message. The maximum reply can be 470 bytes.
0x13	none	Configuration data size too short. The Source Length is less than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x15	none	Configuration data size too large. The Source Length is greater than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x19	none	Data write failure. An error occurred when attempting to write the SMTP server address (attribute 4) to non-volatile memory.
0xFF	0x0100	Error returned by email server; check the Destination string for reason. The email message was not queued for delivery.
	0x0101	SMTP mail server not configured. Attribute 5 was not set with a SMTP server address.
	0x0102	"To:" address not specified. Attribute 1 was not set with a "To:" address AND there is not a "To:" field header in the email body.
	0x0103	"From:" address not specified. Attribute 2 was not set with a "From:" address AND there is not a "From:" field header in the email body.
	0x0104	Unable to connect to SMTP mail server set in Attribute 5. If the mail server address is a hostname, make sure that the device supports DNS, and that a Name Server is configured. If the hostname is not fully qualified, i.e., "mailhost" and not "mailhost.xx.yy.com" then the domain must be configured as "xx.yy.com". Try "ping <mail server address>" to insure the mail server is reachable from your network. Also try "telnet <mail server address> 25" which attempts to initiate a SMTP session with the mail server via telnet over port 25. (If you connect then enter "QUIT").
	0x0105	Communication error with SMTP mail server. An error occurred after the initial connection with the SMTP mail server. See the ASCII text following the error code for more details as to the type of error.
	0x0106	SMTP mail server host name DNS query did not complete. A previous send service request with a host name as the SMTP mail server address did not yet complete. Note that a timeout for a DNS lookup with an invalid host name can take up to 3 minutes. Long timeouts can also occur if a domain name or name server is not configured correctly.

Managing User Accounts and Access Levels

How to Use This Chapter

This chapter describes how to configure user access levels to different information on the module.

For this information:	See page:
User Accounts and Privilege Classes	6-1
Configuring Access Limits for Web Pages	6-2
Creating User Accounts	6-5
Recovering with Unknown Password	6-6

By assigning user accounts with different access levels, you can manage which users have access to change network configuration or have access to view and change data views.

Several pages on the web server module, such as module configuration pages and data views pages, have default access protection. Before accessing these pages, you must authenticate your access by entering a user name and password. The module displays the log-in box when you access these web pages.

IMPORTANT

Once authenticated, you do not have to re-enter a user name or password when accessing subsequent pages. You **must** close your browser to log out.

The default user name is Administrator with no password.

IMPORTANT

It is strongly recommended that you set a password for the default Administrator account.

User Accounts and Privilege Classes

The web server module supports multiple user accounts, each with a user name and password. Each user account is configured for one of these access levels:

- Administrator (all access)
- Write (read and write access)
- Read (read access only)

The access level determines which web pages the user can access. You configure access limits for individual web pages.

Configuring Access Limits for Web Pages

You protect individual web pages and data views on a per URL basis. Each page in the web server module has one of these protection levels:

- Administrator
- Write
- Read

The protection levels are hierarchical. Administrator users can access Write and Read protected pages, and Write users can access Read protected pages.

These pre-defined pages (those web pages that come with the web server module) in the web server module have these default access levels. You can change these access levels, if needed.

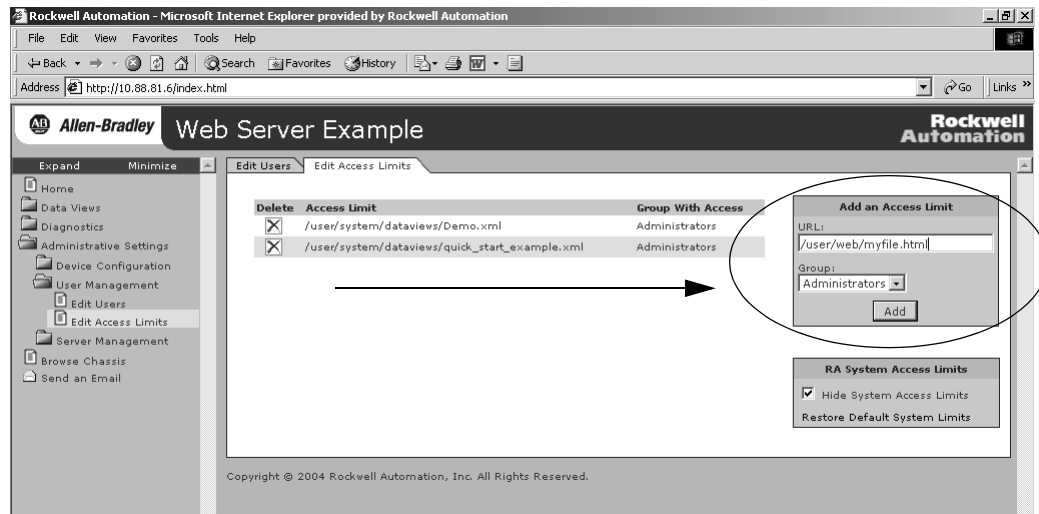
Web Page:	Required Protection Level:
Home page	no protection
Diagnostics pages	
Chassis Browse page	
Data Views with read-only tags	Read protection
Data Views with write tags	Write protection
Data Views with administrator tags	Administrator protection
Device configuration pages	
Server Management page	
User Management page	
Send Email page	

In Data Views, the access limits you specify for a tag applies to the whole data view, not just the tag. If you have multiple tags with different access levels in the same data view, the web server assigns the highest (most access) level to the data view. For more information, see chapter 4.

If you develop custom web pages, you must explicitly specify the access limits for the page if you want access protection. Otherwise, the custom web page will have no access limits. For more information, see chapter 6.

You need Administrator access to modify access limits for web pages. You specify the access limit for a web page from the Administrative Settings → User Management → Edit Access Limits page.

You can change the default access limits for the pre-defined web pages or you can add pages to the protection list, such as custom web pages. The Edit Access Limits page shows the current list of pages the user has selected for protection. The pre-defined pages, though they have default protection, do not show up in the list.


In this field:
Do this:

URL

enter the URL for the web page
80 characters maximum, including slashes.

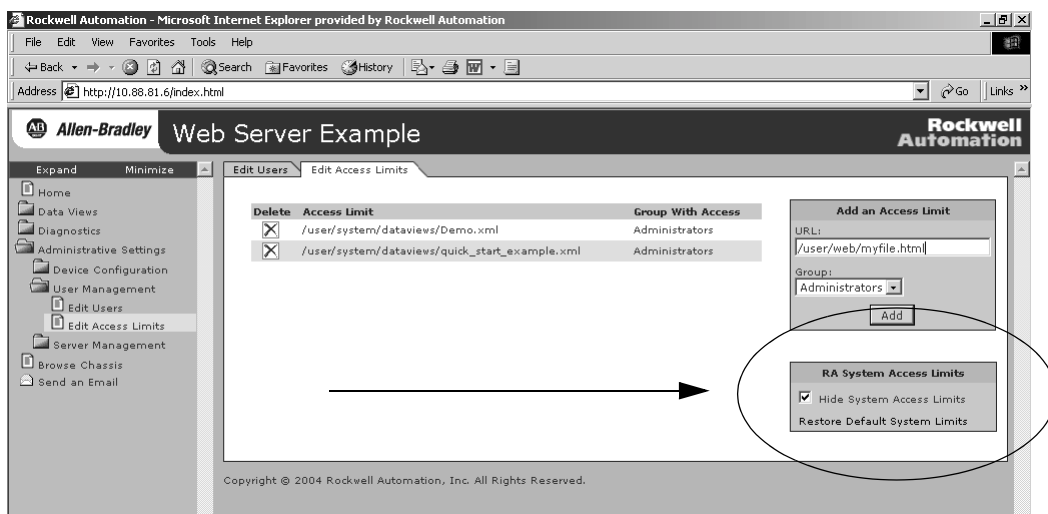
Enter only the relative path of where the page is stored on the web server module, such as "/user/web/mypage.html"

Group

select Administrator, Write, or Read access limit for the specified URL

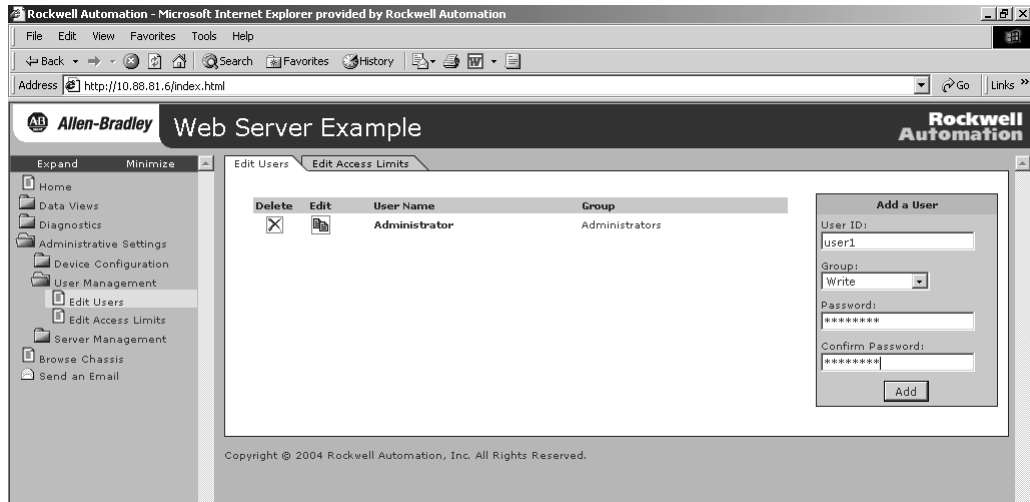
You can configure protection limits for pre-defined pages, as well as for user-supplied pages. You can also apply protection to directories so that all the files in a specified directory have the same access limit. To specify a directory, enter the URL with a frontslash (/) on the end. If you do not enter the frontslash, the protection limit you select is assigned to the web page. To specify an access limit for the entire web site, enter just a frontslash (/).

To see the protection limits for pre-defined web pages and directories, unclick the Hide System Access Limits selection. This also provides access to change the pre-defined protection limits. You can then use the Restore System Access Limits selection to return to the default protection limits.



Creating User Accounts

You need Administrator access to create and modify user accounts. You can create as many as 25 individual accounts. You manage accounts from the Administrative Settings → User Management → Edit Users page.



In this field:	Do this:
User ID	enter the user name for the account 80 characters maximum can contain these characters: A-Z, a-z, 0-9, underscore (_), and dash (-)
Group	select Administrator, Write, or Read access for the user account
Password	enter the password for the account 80 characters maximum
Confirm Password	re-enter the same password for the account

IMPORTANT

If you use Internet Explorer, the number of characters allowed for a user ID or password depends on how many characters “fit in the box.” Larger characters (such as “W”) take more room and reduce the total number of allowed characters.

This limitation does not apply if you use Netscape, Mozilla, or some other browser.

Recovering with Unknown Password

There are no “back door” accounts or passwords in the event that the you forget the web server module’s passwords or inadvertently delete all the Administrator accounts.

To recover a web server module with unknown passwords, you must use ControlFlash to restore the web server’s flash file system back to the factory default. This operation deletes all user accounts, data views, and user-loaded web pages. Contact technical support for the appropriate recovery script and binary file.

Accessing Files in the Web Server Module

How to Use This Chapter

This chapter describes how to use FTP to access the file system on the web server module.

For this information:	See page:
Accessing the Web Server's File System	7-1
Backing Up the File System on the Web Server Module	7-5

You use FTP access to store custom web pages and applications on the web server module.

Accessing the Web Server's File System

The web server module has a flash file system that stores web pages and data views. There is approximately 5 Mbytes of space in the flash file system to store data views and custom web pages. Some predefined directories exist to store specific types of data:

Use this directory:	For:
/user	<p>This is the highest directory level you can access on the web server module. It contains two subdirectories:</p> <ul style="list-style-type: none"> • /web for you to store your custom-created pages or other files • /system to store configuration files and data views <p>You can only access this directory (via FTP) when the web server module is in backup/restore mode.</p>
/user/web	<p>Use this directory to store your custom-created pages or own files. Standard FTP provides access to this directory; the module does not need to be in any special mode.</p>
/user/system	<p>This predefined directory contains two predefined directories:</p> <ul style="list-style-type: none"> • /configuration to store network and module configuration files • /dataviews to store data view files <p>You can only access this directory (via FTP) when the web server module is in backup/restore mode.</p>
/user/system/configuration	<p>This predefined directory contains network and module configuration files in an XML format. You can only access this directory (via FTP) when the web server module is in backup/restore mode.</p>

Use this directory:	For:
/user/system/dataviews	This predefined directory contains data view files in an XML format. You can only access this directory (via FTP) when the web server module is in backup/restore mode.
/schema	This directory contains dataview.xsd and CIPDataTypes.xsd schema files for validating data views. You can only access these .xsd files with a web browser: http://ip_address/schema/dataview.xsd or http://ip_address/schema/CIPDataTypes.xsd
/dataview	This predefined directory contains dataview.xml which is the external XSL stylesheet for data views. You can only access this .xml file with a web browser: http://ip_address/dataview/dataview.xml

There are no restrictions on the type of files you can copy to the web server module. You are only restricted by the amount of memory available.

You access this file system using any standard FTP client. By default, FTP is disabled for the web server module. You enable FTP on the Administrative Settings → Device Configuration → Device Services page (see page 2-13).

Connecting to the web server module

IMPORTANT

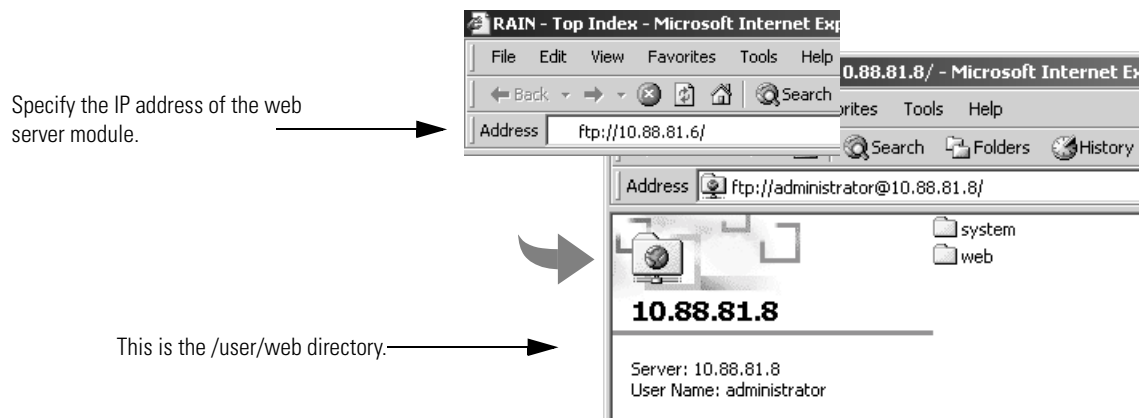
FTP access must be enabled for the web server module before you can use FTP to access the module. You enable FTP on the Administrative Settings → Device Configuration → Device Services page (see page 2-13).

To establish a FTP connection to the web server module, you need Administrator access to the module. You will also need to enter your user name and password.

IMPORTANT

The password is transmitted over the network in plain text. It is recommended that you create a temporary Administrator account for when you use FTP to access to the web server module. Also, leave FTP disabled except for when actively copying files to or from the web server module.

To connect to the web server module via FTP, specify the IP address of the module.



Once connected to the web server module, you have access to the /user/web directory. You can create subdirectories, but you cannot access any directories higher than this user directory (which contain pre-defined web pages and files).

The web server module supports four FTP sessions (four different users with simultaneous FTP access to the module). If you use Internet Explorer to FTP to the web server module, there are only two FTP sessions available.

When the web server module is in backup/restore mode, an FTP session points to a different directory than when the web server module is not in backup/restore mode. If you have a standard FTP session open to the web server module, close this before placing the module in backup/restore mode. Once in backup/restore mode, the previously open FTP session will no longer point to the correct directory and files.

File names and types

File names can have no more than 80 characters. In addition, the complete path for any file in the file system can have no more than 80 characters. For example, the user-created web pages go in the /user/web folder, which is actually in the file system as “/root/user/web” (already 14 characters). Any path underneath this must have 65 or less characters, including slashes, dots, and extensions.

File names are case-insensitive and can contain any characters, including spaces, except for the following characters: ? “ / \ < > * | :

When user files are accessed via HTTP, certain file extensions result in specific values returned in the Content-Type field. Content-Types for the more commonly-used files are:

File Extension:	Content-Type:
.htm	text/html
.html	text/html
.asp	text/html
.gif	image/gif
.jpg	image/jpeg
.css	text/cascading style sheet
.txt	text/plain
.js	application/x-javascript
.exe	application/binary
.z	application/compressed
.gz	application/gzip
.bin	application/octet-stream
.oda	application/oda
.pdf	application/pdf
.ai	application/postscript
.eps	application/postscript
.ps	application/postscript
.xml	text/xml
.xsl	text/xml

In addition, files with a “.asp” extension are processed by the web server as Active Server Pages. For more information on creating custom web pages, see chapter 8.

Backing Up the File System on the Web Server Module

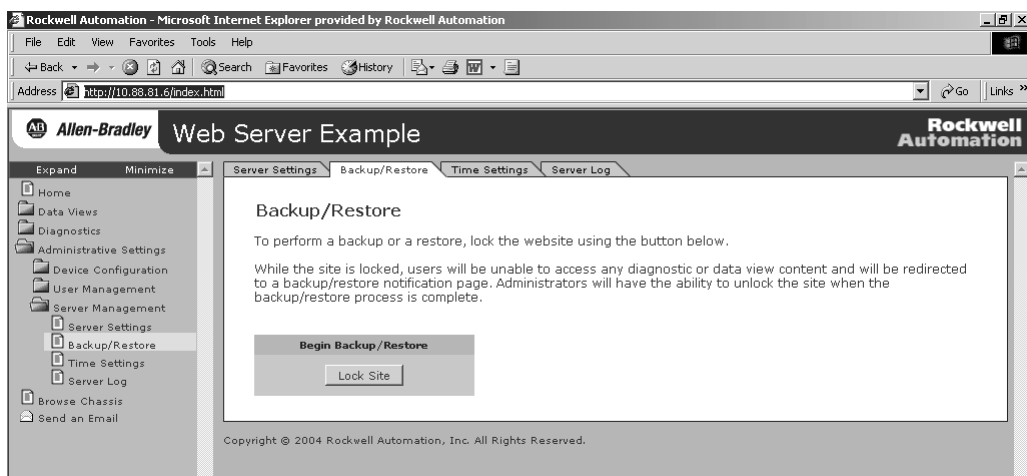
There are a several items that are stored on the web server module that you might want to archive in a backup copy:

- user accounts and passwords
- list of URLs the user has selected for read, write, or administrator protection
- data views
- custom web pages
- module configuration data

To back up these files, you use standard FTP to access the web server module and then copy or restore files. Using FTP lets you copy files to or from the web server module without interrupting operation of the module. FTP also lets your standard FTP-capable clients, such as Internet Explorer or WinZip to copy the system files.

FTP access during a backup/restore procedure differs from normal FTP access in that you get access to the /user directory, which contains the /web directory that is accessible during normal FTP access. This lets you backup pre-defined web pages and data views, as well as any custom files you have copied to the web server module.

The Administrative Settings → Server Management → Backup/Restore lets you lock access to the module while you back up or restore files.



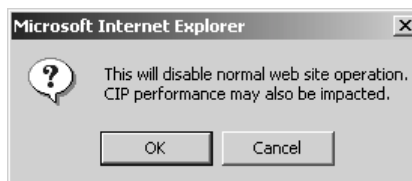
Locking the site prevents other users from HTTP access to the web server module. Locking the site does not affect bridge functionality through the web server module, however, restoring files to the web server module that modify network or module configuration settings (such as IP address) can affect bridge functionality.

Backing up files

You must have Administrator access to backup files. To back up files that reside on the web server module:

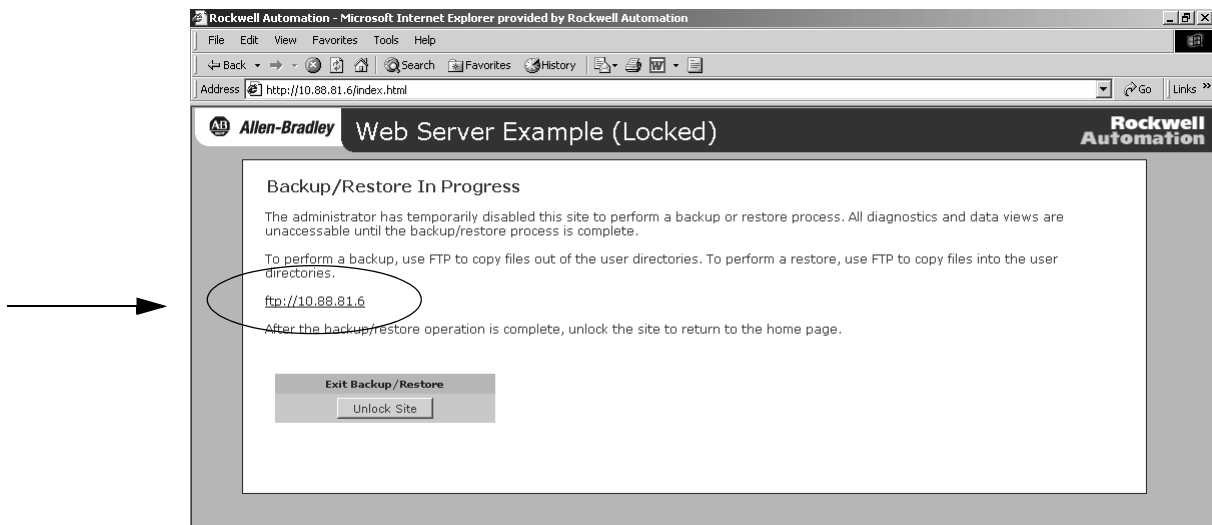
1. Click the Lock Site button on the Administrative Settings → Server Management → Backup/Restore page to lock access to the web server module.

You must lock access to the web server module to gain FTP access to the /user directory. The web server module asks to confirm that you want to lock the site.



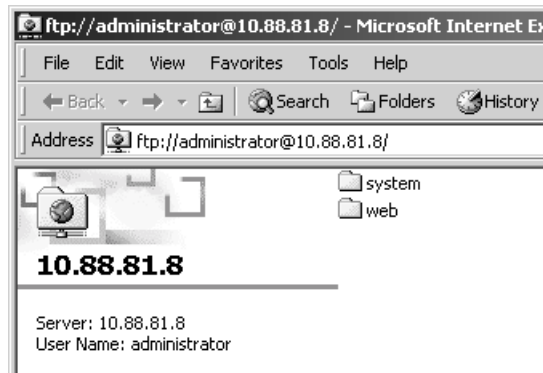
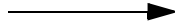
This automatically enables FTP access to the web server module.

2. Click the link to the web server module. This appears on the locked backup/restore page.



The module will require that you enter a valid user name and password that allows administrator access. Once authenticated, you have access to the /user directory. This is one directory higher than the /user/web directory available during normal FTP operations.

This is the /user directory.



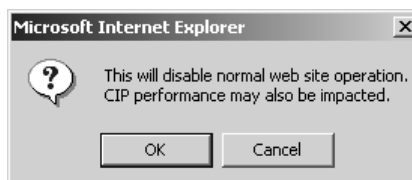
3. Select the files you want to back up.
4. Copy the selected files to a directory on your PC.
5. Unlock access to the module on the Administrative Settings → User Management → Backup/Restore page.

Restoring files

You must have Administrator access to restore files. To restore files from your PC to a web server module:

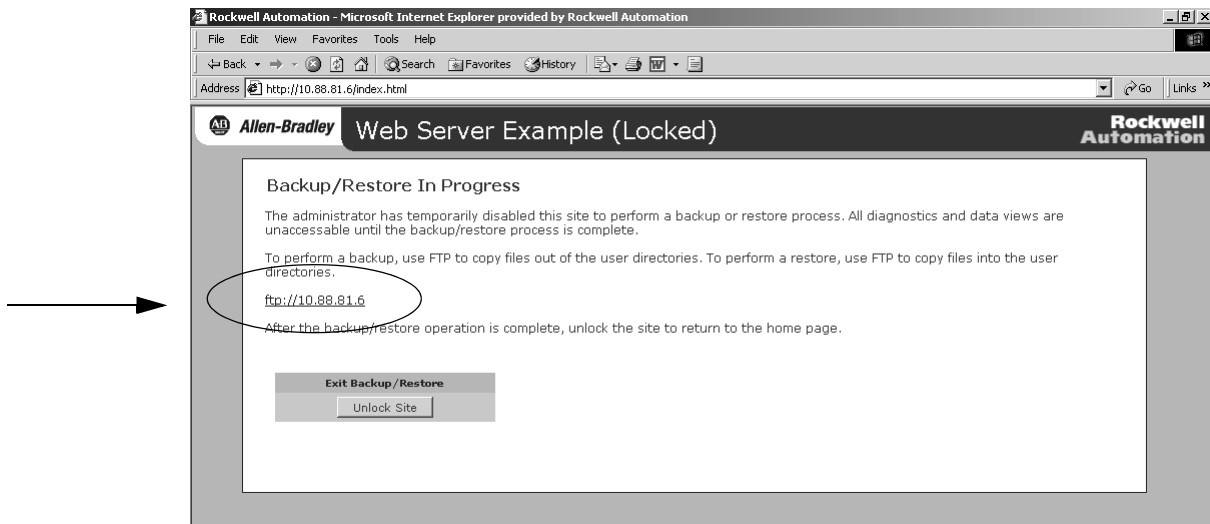
1. Click the Lock Site button on the Administrative Settings → Server Management → Backup/Restore page to lock access to the web server module.

You must lock access to the web server module to gain FTP access to the /user directory. The web server module asks to confirm that you want to lock the site.

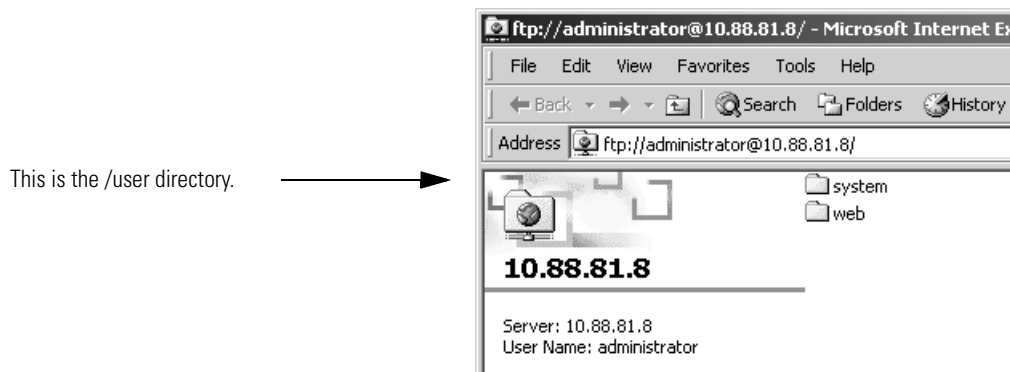


This automatically enables FTP access to the web server module.

2. Click the link to the web server module. This appears on the locked backup/restore page.



The module will require that you enter a valid user name and password that allows administrator access. Once authenticated, you have access to the /user directory. This is one directory higher than the /user/web directory available during normal FTP operations.



3. Select the files from your PC that you want to restore to the web server module.
4. Copy the selected files to a directory on the web server module.
5. Unlock access to the module on the Administrative Settings → User Management → Backup/Restore page.

Creating Custom Web Pages

How to Use This Chapter

This chapter describes how to use ASP functions in custom web pages and how to load custom web pages into the web server module.

For this information:	See page:
Overview	8-1
Developing a Custom Web Page	8-3
ASP Function Calls	8-4
Javascript Libraries	8-9
Web Page Forms and POST Handlers	8-12

IMPORTANT

The web server module provides access to tags within a local controller via a web browser. However, the web server module is not recommended for use as a real-time HMI or operator interface.

The pre-defined web pages that come with the module provide one method of accessing these tags. The ability to load and run custom web pages gives you the flexibility to design web pages that better fit your application. For example, you can define a web page with standard web content, such as a company logo, contact information, and links to other web pages. Add the ASP functions to display live controller data.

Overview

Use your own editor or application, to develop the appropriate HTML and ASP files for your custom pages. Once these files are ready, you have to copy the files to the web server module and configure the use of those pages as needed. The following steps outline the process of developing custom web pages and getting them ready to use:

1. Develop the appropriate HTML and ASP files.

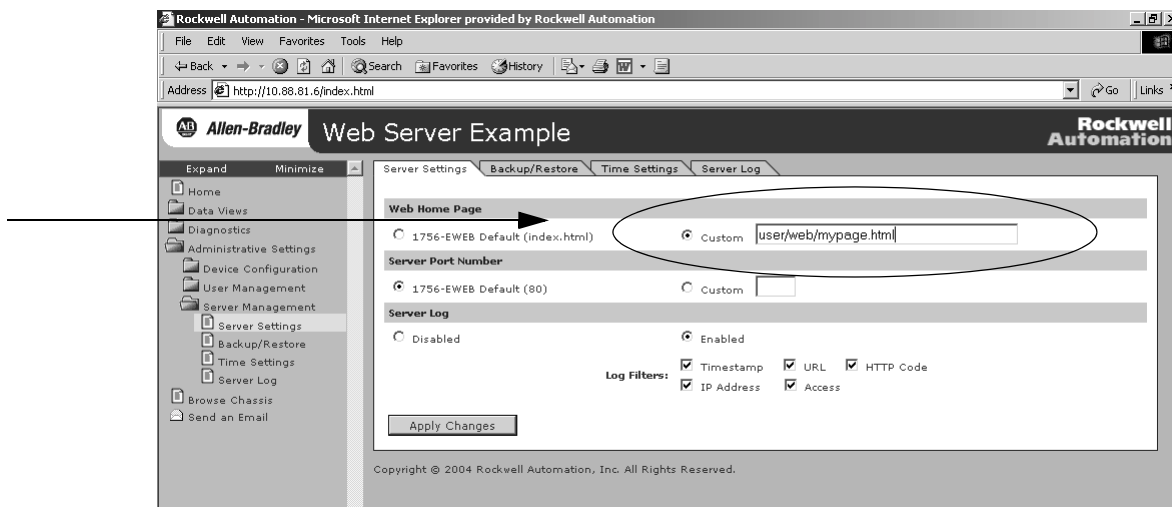
See Developing a Custom Web Page on page 8-3 for information about the available ASP functions and other features for displaying and changing controller data.

2. Use FTP to copy the custom pages to the web server module.

Copy your files into the /user/web directory. See chapter 7 for more information.

3. Decide whether or not to make your custom page the default Home page for the web server module.

Use the Administrative Settings → Server Management Server → Server Settings page. When you click the Custom button, the web server module automatically fills in the “/user/web” portion of the location of the custom file. Enter in any further directories and the file name. Click Apply Changes.



4. Determine whether you want to set access limits, such as Administrator, Write, or Read, on the custom page.

Use the Administrative Settings → User Management → Edit Access Limits page to set access limits on specific pages. See chapter 6 for more information.

Accessing custom web pages

Once a custom web page is copied into the web server module, you can display that page from your web browser:

If you custom web page is:	Then, in your web browser:
configured as the default home	<p>enter the IP address of the web server module.</p> <p>The web browser accesses the defined web page for the web server module.</p> <p>If you specify a custom page as the default home page, you can still access the pre-defined home page by entering: <code>http://ip_address/index.html</code></p>
not configured as the default home page, but is copied into the <code>/user/web</code> directory	<p>enter: <code>http://ip_address/user/web/my_file.html</code></p> <p>The web browser displays any custom file if you specify the complete URL to that file.</p>

Developing a Custom Web Page

Custom web pages can contain standard web content, such as a company logo, contact information, and links to other web pages. In addition, you can use ASP functions and other features that come with the web server module to display and manipulate live controller data.

An ASP file is essentially the same as an HTML file, with embedded scripting constructs supported by ASP. An ASP file has the file extension “.asp”. When an ASP file is requested by a browser, the web server interprets the ASP file and executes the ASP-specific scripts. The browser sees only the resulting HTML, not the embedded ASP scripts.

The web server module provides:

Use	To:	See page:
ASP function calls	display controller data	8-4
Javascript libraries	manipulate the controller data returned by ASP function calls	8-9
Web page forms and POST handlers	change controller data	8-12

ASP Function Calls

The web server module provides these ASP functions:

To:	Use this ASP function:	See page:
Read controller tags	ReadLogixTag	8-4
	ReadLogixTagUnconnected	8-5
Read CIP data	CIPMessage	8-5
	CIPMessageUnconnected	8-6
Retrieve information about the module	GetSetting	8-7

Read Controller Tags

There are two ASP functions you can use to read controller tags:

If you want:	Use this ASP function:	See page:
connected messaging	ReadLogixTag	8-4
unconnected messaging	ReadLogixTagUnconnected	8-5

Function:

`ReadLogixTag(path, tagname, tagtype)`

The `ReadLogixTag` function connects to a controller at a specified *path*, retrieves the value of a tag with a name *tagname* and of *tagtype*.

Parameters:

Parameter:	Description:				
<i>path</i>	The <i>path</i> is a CIP path, with no spaces, and segments separated by commas. For example: <table border="1"> <thead> <tr> <th>Path Example:</th><th>Description:</th></tr> </thead> <tbody> <tr> <td>1,3</td><td>1 = backplane 3 = slot 3</td></tr> </tbody> </table>	Path Example:	Description:	1,3	1 = backplane 3 = slot 3
Path Example:	Description:				
1,3	1 = backplane 3 = slot 3				
<i>tagname</i>	The <i>tagname</i> identifies a controller-scoped tag.				
<i>tagtype</i>	The <i>tagtype</i> must be an atomic type (BOOL, SINT, INT, DINT, REAL) or string. Use standard dot notation to specify a member of an array or a user-defined structure (e.g. "timer1.ACC" for the accumulator of a timer tag named "timer1").				

Example: For example, retrieve a DINT tag named “my_dint_tag” from a controller in slot 3 of the local chassis:

```
<p> "my_dint_tag" value:
<% ReadLogixTag("1,3", "my_dint_tag", "DINT"); %>
</p>
```

Function:

`ReadLogixTagUnconnected(path, tagname, tagtype)`

The `ReadLogixTagUnconnected` function performs the same task as `ReadLogixTag` function and uses the same parameters. This function, however, retrieves the tag value through unconnected messaging, rather than connected messaging.

Read CIP Data

There are two ASP functions you can use to read CIP data:

If you want:	Use this ASP function:	See page:
connected messaging	<code>CIPMessage</code>	8-5
unconnected messaging	<code>CIPMessageUnconnected</code>	8-6

Function:

`CIPMessage(path, service, class, instance, attribute, member, data, returntype)`

The `CIPMessage` function performs the CIP service specified by the *service* parameter on an object or object instance specified by the *class*, *instance*, *attribute*, and *member* parameters, using the data passed in the *data* parameter (if necessary) and returning the value with a type specified by the *returntype* parameter (if appropriate).

Parameters:

Parameter:	Description:
<i>path</i>	The <i>path</i> is a CIP path, with no spaces, and segments separated by commas. This is the same as the <i>path</i> parameter in the read-controller-tag functions.
<i>service</i>	The <i>service</i> identifies the CIP function to perform.
<i>class</i>	The class, instance, attribute, and member parameters identify the object for the service.
<i>instance</i>	
<i>attribute</i>	
<i>member</i>	
<i>data</i>	The data to be passed. If no data is passed in the request, set the <i>data</i> parameter to an empty string.
<i>returntype</i>	The <i>returntype</i> parameter is the same as the <i>tagtype</i> parameter used in the read-controller-tag functions, except when a <i>returntype</i> of "STRING" is specified, the data returned from the service is written to the browser as a space-delimited string of hex digits.

Example: For example, a GET_ATTRIBUTE_ALL to the identity object of a device in slot 1 of the local chassis:

```
<% CIPMessage("1,1", 1, 1, 1, 0, 0,"","STRING"); %>
```

This ASP call returns a string similar to:

```
01 00 0E 00 03 00 0C 0C 70 30 63 2E 08 00 1D 31 37 35
36 2D 4C 31 2F 41 20 31 37 35 36 2D 4D 30 2F 30 20 41
52 47 5F 31 32 5F 33 38
```

While the value returned by specifying the "STRING" data type may not be very useful to display in the browser by itself, some built-in Javascript libraries help the ASP developer parse and use the information contained in these string structures. See page 8-9 for more information on the Javascript libraries.

Function:

```
CIPMessageUnconnected(path, service, class, instance, attribute, member, data,  
returntype)
```

The CIPMessageUnconnected function performs the same task as the CIPMessage function and expects exactly the same parameters. This function, however, performs the specified service through unconnected messaging, rather than connected messaging.

Updating control system data

You can use the `CIPMessage` and `CIPMessageUnconnected` functions to perform any CIP service, such as a `SET_ATTRIBUTE_SINGLE`, `RESET`, or any other type of write or update service to an object or object instance. However, since ASP pages are parsed and executed server-side, it is impossible to send any dynamic data to these services.

As an example, consider a page which performs a reset to the identity object of a device in slot 1, then immediately redirects the browser to the main user page:

```
<HTML>
<HEAD>
<% CIPMessageUnconnected("1,1", 5, 1, 1, 0, 0, "", "STRING"); %>
<META HTTP-EQUIV="refresh" content="0;URL=/user/index.html">
</HEAD>
<BODY></BODY>
</HTML>
```

Retrieve Information about the Web Server Module

You can retrieve specific information about the web server module.

Function:

`GetSetting(settingname)`

The `GetSetting` function retrieves a specific piece of information about the web server module itself and writes this data to the browser.

Parameters:

Parameter:	Description:
<i>settingname</i>	The <i>settingname</i> specifies the piece of information to retrieve.

where *settingname* can be:

Value:		Description:
"uptime"	"device_status"	These fields as displayed on the pre-defined home page.
"firmware_revision"	"firmware_version"	
"serial_number"		See page 1-4 for more information.
"name"	"description"	These fields as configured on the Administrative Settings → Device Configuration → Device Identity page.
"location"	"contact_1"	
"contact_2"		See page 3-2 for more information.
"ip_address"	"subnet_mask"	These fields as configured on the Administrative Settings → Device Configuration → Network Configuration page.
"default_gateway"	"primary_name_server"	
"secondary_name_server"	"domain_name"	
"host_name"	"obtain_configuration"	See page 3-3 for more information.
"dns_enable"	"smtp_server"	
"smtp_username"	"smtp_authEnabled"	
"ethernet_address"	"autonegotiation"	
"port_speed"	"duplex_mode"	
"cpu_utilization"	"file_sys_utilization"	The diagnostic information on the Diagnostics → Diagnostic Overview page.
"server_errors"	"server_redirects"	
"server_timeouts"	"server_access_violations"	
"server_page_hits"	"server_form_hits"	See page 9-2 for more information.
"server_total_hits"	"user_free_space"	
"server_data_views"	"server_total_tags"	
"tcp_conns"	"tcp_conn_limit"	
"tcp_max_conns"	"cip_conns"	
"cip_conn_limit"	"cip_max_conns"	
"cip_conn_opens"	"cip_conn_open_errors"	
"cip_conn_closes"	"cip_conn_close_errors"	
"cip_conn_timeouts"	"cip_msg_sent"	
"cip_msg_rcv"	"cip_ucmm_sent"	
"cip_ucmm_rcv"		
"if_in_octets"	"if_in_ucast"	The diagnostic information on the Diagnostics → Ethernet Statistics page.
"if_in_nucast"	"if_in_discards"	
"if_in_errors"	"if_in_unknown_protos"	
"if_out_octets"	"if_out_ucast"	See page 9-7 for more information.
"if_out_nucast"	"if_out_discards"	
"if_out_errors"	"media_alignment_errors"	
"media_fcs_errors"	"media_single_collisions"	
"media_multiple_collisions"	"media_sqe_test_errors"	
"media_deferred_trans"	"media_late_collisions"	
"media_excessive_collisions"	"media_mac_trans_errors"	
"media_carrier_sense_errors"	"media_frame_too_long"	
"media_mac_receive_errors"		
"time"		time in seconds since January 1, 1970
"asc_local_time"		ASCII string of time and date
		These values display the time the web page is accessed.

Javascript Libraries

The web server module provides built-in Javascript functions designed to help manage control system data in custom web pages.

Use this Javascript library:	To:	See page:
conversion.js	convert values in the string returned from CIPMessage and CIPMessageUnconnected functions using the STRING data type into atomic, numeric or string values.	8-9
XMLObjectLoaderLib.js	convert the string returned from a GET_ATTRIBUTE_ALL service into a Javascript object which lets you refer to the object or instance attributes by name.	8-10

Javascript library: conversion.js

The conversion.js library contains six functions. To include this library in your custom web page, include this line:

```
<script type="text/javascript" src="/scripts/conversion.js">
</script>
```

The functions in the conversion.js library are:

Function:	Description:
parseStruct(structStr, type, stringlength)	<p>The parseStruct function is the main function, where:</p> <p><i>structStr</i> contains the string returned from the ASP call, or a substring thereof, beginning at the hex number which starts the value to be retrieved from the string.</p> <p><i>type</i> indicates the data type to be parsed from the string. It should be one of "SINT", "USINT", "INT", "UINT", "WORD", "DINT", "UDINT", "DWORD", or "STRING".</p> <p><i>stringlength</i> (optional) only used when the type to be parsed is a "STRING". In this case, it identifies how many digits of the input string to parse into ASCII characters.</p>

For example, to retrieve the device name from a device in slot 1 of the local chassis:

```
var idobj = "<% CIPMessage("1,1", 1, 1, 1, 0,
0, "0", "STRING"); %>";
var namelen = parseStruct(idobj.substring(42,
idobj.length), "USINT");
var name = parseStruct(idobj.substring(45,
idobj.length), "STRING", namelen);
```

The Device Name attribute of the identity object instance starts 14 bytes into the GET_ATTRIBUTE_ALL response. Each byte of the response in the "STRING" return style takes 3 bytes (2 hex digits and a space), so the Device Name attribute begins at the 42nd character of the string. The first byte of the string contains the number of characters in the string. After parsing this string length, pass the length on to the next function, which parses the actual string beginning at the 15th byte (45th character) of the string.

Function:	Description:
<code>decToHex (decimalnumber)</code>	The <code>decToHex</code> function takes an unsigned decimal number as a parameter and returns a string representing <i>decimalnumber</i> in hexadecimal notation. The return value does not have a preceding "0x" and the returned string is always 8 characters long, with leading zeros when necessary.
<code>hexToDec (hexnumber)</code>	The <code>hexToDec</code> function takes a string containing a hexadecimal number with no preceding "0x" as a parameter, and returns a decimal number with the value of <i>hexnumber</i> .
<code>decToOct (dintnumber)</code>	The <code>decToOct</code> function takes an unsigned decimal number as a parameter and returns an 11-character string representing <i>dintnumber</i> in octal notation. The return value does not have a preceding "0" and the returned string is always 11 characters long, with leading zeros when necessary.
<code>decToBin (decimalnumber)</code>	The <code>decToBin</code> function takes an unsigned decimal number as a parameter and returns a string representing <i>decimalnumber</i> in binary notation. The return value does not have a prefix and breaks the resulting binary string into groups of four characters.

Javascript library: XMLObjectLoaderLib.js

This library uses the `conversion.js` library. To include these libraries in your custom web page, include these lines:

```
<script type="text/javascript" src="/scripts/conversion.js">
</script>
<script type="text/javascript" src="/scripts/XMLObjectLoaderLib.js">
</script>
```

The `XMLObjectLoaderLib` library requires an input file in XML format which describes the CIP object being returned from a `GET_ATTRIBUTE_ALL` service. The following example shows the format of this input file:

```
<object name="IdentityInstance">
  <attribute index="1" name="VendorID" type="UINT"/>
  <attribute index="2" name="DeviceType" type="UINT"/>
  <attribute index="3" name="ProductCode" type="UINT"/>
  <object index="4" name="Revision">
    <attribute index="1" name="MajorRevision"
type="USINT"/>
    <attribute index="2" name="MinorRevision"
type="USINT"/>
  </object>
  <attribute index="5" name="Status" type="WORD"/>
  <attribute index="6" name="SerialNumber" type="DWORD"/>
  <attribute index="7" name="ProductName" type="STRING"/>
</object>
```


This description format is recursive - structures within the object can be described by declaring an “object” element, then describing the members of the structure under consideration. The names of the elements are used to construct the Javascript object, and the members of the Javascript object have names that match the “name” attributes in the description file. The “index” attributes describe the order in which these elements occur in the object; each attribute and object except the root object must contain an index attribute.

The functions in the XMLObjectLoaderLib.js library are:

Function:	Description:
<code>XMLObjectLoader()</code>	<p>This routine initializes the <code>XMLObjectLoader</code> object used to parse the CIP return data. Set it to a declared variable in the Javascript. For example:</p> <pre>var xol = new XMLObjectLoader();</pre>
<code>LoadObject(objectname, xmlfilename, objectstruct)</code>	<p>The <code>LoadObject</code> function is a member function of the <code>XMLObjectLoader</code> object that parses the CIP object return string passed in <i>objectstruct</i>, using the XML description file <i>xmlfilename</i>, and placing the result in the <i>objectname</i> variable, where:</p> <p><i>objectname</i> global variable that contains the result, accessible to all scripts within the page.</p> <p><i>xmlfilename</i> absolute or relative path to the XML file containing the object description</p> <p><i>objectstruct</i> CIP object return string</p>

For example, the following code is a snippet from a web page which displays information about the identity object of a device in slot 1 of the local chassis:

```
<script type="text/javascript" src="/scripts/conversion.js"></script>
<script type="text/javascript" src="/scripts/XMLObjectLoaderLib.js"></script>
<script type="text/javascript" language="javascript"> var idobj;

function loadPage() {
    document.getElementById("vendor").innerText = "" + idobj.VendorID;
    document.getElementById("type").innerText = "" + idobj.DeviceType;
    document.getElementById("code").innerText = "" + idobj.ProductCode;
    document.getElementById("rev").innerText = idobj.Revision.MajorRevision + "."
+
    idobj.Revision.MinorRevision;

    document.getElementById("status").innerText = "0x" + decToHex(idobj.Status);
    document.getElementById("serial").innerText = "0x" +
decToHex(idobj.SerialNumber);
    document.getElementById("prodname").innerText = idobj.ProductName;
}

function loadme() {
    var id = "<% CIPMessage("1,1", 1, 1, 1, 0, 0,"0","STRING"); %>"
    var xl = new XMLObjectLoader();
    xl.LoadObject("idobj", "/CIPXMLobj/CN_01_Identity.xml", id);
    loadPage();
</script>

<body onload="loadme();" ...>
etc...
```

Web Page Forms and POST Handlers

Any web page that displays controller data can include a way to change that data by creating a form on the web page. Include this statement in your web page file:

ACTION="*change_method*" METHOD="POST"

where *change_method* is one of these:

Use This Method:	To:	See page:
/user/system/dataviews/ <i>filename.xml</i>	change all data in a data view or only specific data in a data view	8-13
/rokform/WriteLogixTags	change any data in a controller, whether it is in a data view or not	8-17
/rokform/ReadLogixTags	enter a tag name and receive its current value	8-19
/rokform/CIPMessage	send an instant CIP message request	8-20

`ACTION="/user/system/dataviews/filename.xml"`

This method follows the same user access restrictions as the user requesting the data view. The user can only post the tags that appear on the data view and the user must have the same access level as configured for the data view. You can post:

- the same XML that you got in the requested file with the desired data values changed

or

- only the controller tags you want to change

For example, this code:

```
<html>
<head>
<title>
Data View Javascript Demo Page
</title>

<script type="text/javascript">

var xmlDoc;
var xmlhttp;

var DATAVIEW_URI = "/user/system/dataviews/Sample.xml";

var DATAVIEW_NAMESPACE = "http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/";
var DATATYPES_NAMESPACE = "http://www.rockwellautomation.com/technologies/data_access/data_types/1.0/";

var ARR_ACTIVEX = ["MSXML4.DOMDocument", "MSXML3.DOMDocument", "MSXML2.DOMDocument", "MSXML.DOMDocument",
"Microsoft.XmlDom"];
var STR_ACTIVEX = "";

var isIE = navigator.userAgent.toLowerCase().indexOf("msie") > -1;

if (isIE) {
    var bFound = false;

    for (var i=0; i < ARR_ACTIVEX.length && !bFound; i++) {
        try {
            var objXML = new ActiveXObject(ARR_ACTIVEX[i]);
            STR_ACTIVEX = ARR_ACTIVEX[i];
            bFound = true
        } catch (objException) {}
    }
}
```

```
if (!bFound) {
    /* No DOM found, so throw a message and go back */
    alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with XML support. Please
upgrade your browser or load the appropriate patches to support XML.");
    history.go(-1);
}
} else {
    /* Not IE, so throw a message and go back */
    alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with XML support. Please
upgrade your browser or load the appropriate patches to support XML.");
    history.go(-1);
}

xmlDoc = new ActiveXObject(STR_ACTIVEX)
xmlDoc.async = false;
xmlDoc.setProperty("SelectionLanguage", "XPath");
if (STR_ACTIVEX.indexOf(".DOMDocument") != -1) {
    xmlhttp = new ActiveXObject(STR_ACTIVEX.substring(0, STR_ACTIVEX.indexOf(".DOMDocument"))) +
".XMLHTTP");
} else {
    alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with XML support. Please
upgrade your browser or load the appropriate patches to support XML.");
    history.go(-1);
}

function refresh_value() {
    xmlDoc.load(DATAVIEW_URI);
    xmlDoc.setProperty("SelectionNamespaces", "xmlns:dv='" + DATAVIEW_NAMESPACE + "'");
    document.getElementById("tagvalue").value =
xmlDoc.selectSingleNode("/dv:view/dv:tag[@name='TEST']/dv:value").childNodes[0].nodeValue;
}

function update_value() {
    xmlDoc.selectSingleNode("/dv:view/dv:tag[@name='TEST']/dv:value").childNodes[0].nodeValue =
document.getElementById("tagvalue").value;
    xmlhttp.Open("POST", DATAVIEW_URI, false);
    xmlhttp.Send(xmlDoc);
    refresh_value();
}

</script>

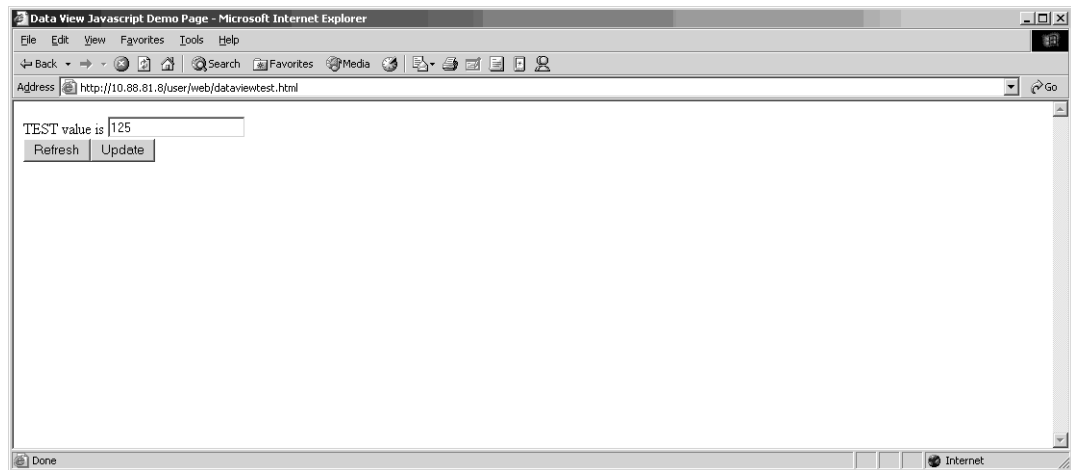
</head>

<body onLoad="refresh_value();">

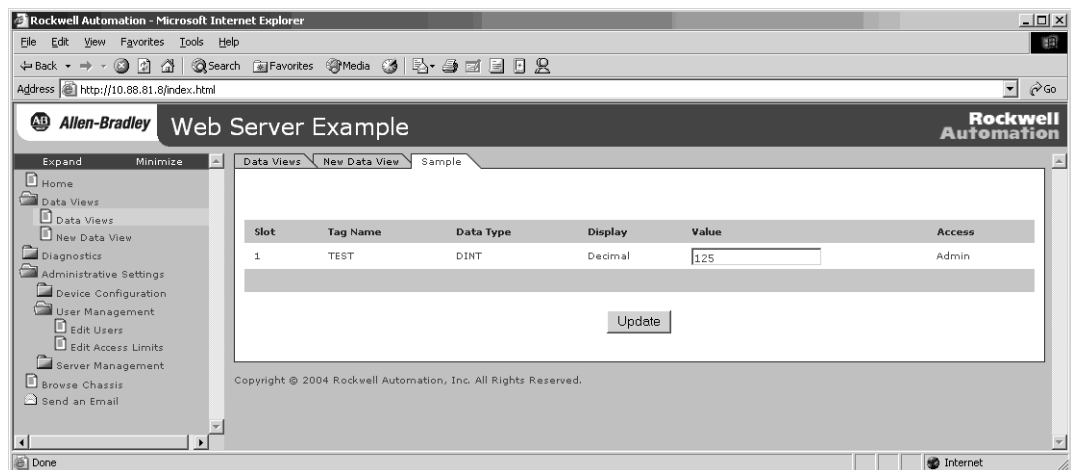
TEST value is <input type="text" name="tagvalue" id="tagvalue" /><br/>
<input type="button" onClick="refresh_value();" value="Refresh"/><input type="button"
onClick="update_value();" value="Update"/>

</body>
</html>
```

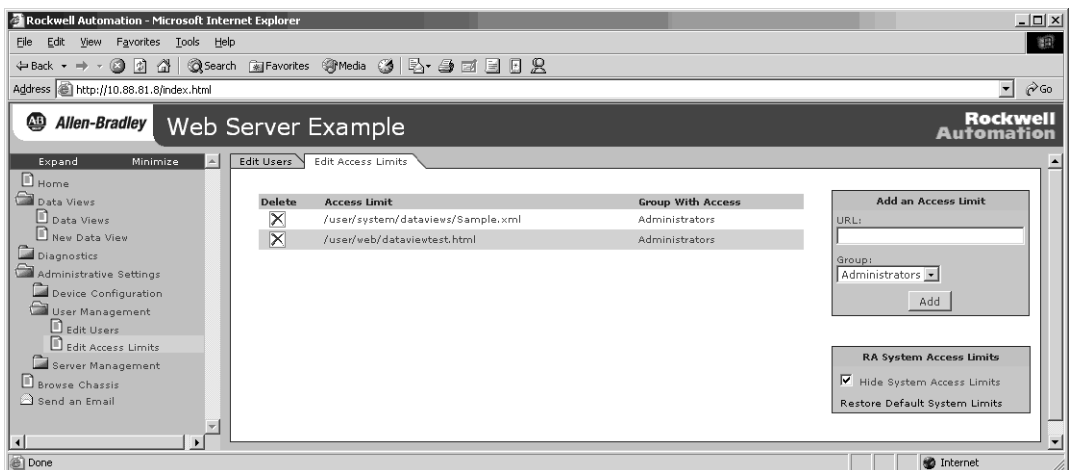
displays this screen:



In the web server module, the data view looks like this:



This example also requires a custom access level.



This method (updating the data view) is geared toward programmatic updates. When an external application that loads a data view wants to change values in that data view, the application can post a modified version of the data view to a special URL on the web server module:

1. Load the XML data view from its URL (e.g. `"/user/web/system/dataviews/myview.xml"`).
2. Programmatically modify the values of any tags.
3. Remove any tags that were not modified from the XML document.
4. Either post the XML file as a file attachment or include its contents in a parameter named `"xml"` posted to the URL of the data view itself.
5. Include a redirect parameter to direct a user to a page showing an up-to-date copy of the data view being modified.

Consider:

- Only tags marked as `"Write"` or `"Administrator"` can be changed. All changes to the values of `"Read"` tags are ignored.
- When the program posts an update to the data view, it must present the same user authentication that was necessary to initially load the data view.
- Any errors in the XML, whether `"valid"` (in XML) manipulations of the original document or malformed XML content, return an HTTP error to the client program.

ACTION="/rokform/WriteLogixTags"

This method is a web POST form handler that requires Administrator access. With the correct access level, a user can access and post to any tag in the controller, not just tags in a data view. You can lower the access restriction, but this opens access to all controller data to more users.

The form must include the following hidden inputs:

Hidden Input:	Description:
"redirect"	A URL where the browser should be redirected after the tag values have been written. Usually equal to the URL of the current page.
"numtags"	The total number of controller tags on the page.

For each controller tag, define these input fields (where "@" represents the index of the tag on the page, e.g. "1" for the first tag, "2" for the second, etc.):

Input:	Description:
"t_@_tagname"	(hidden) The name of the controller tag to be updated. Should be the same as the name used in a data view, or in a call to ReadLogixTag.
"t_@_slot"	(hidden) The slot number of the controller containing the tag to be written.
"t_@_type"	(hidden) The data type of the tag to be written. Should be one of the types defined in "/schema/CIPDataTypes.xsd" on the web server module (e.g. "DINT").
"t_@_display"	(hidden) The display format of the tag to be written. Should be "Decimal", "Hexadecimal", "Octal", "Binary", or "String". Both ReadLogixTag and ReadLogixTagUnconnected return numbers in decimal format. Unless you convert these numbers into another format, use "Decimal" for these values.
"t_@_changed"	(hidden) A "0" or a "1" to indicate whether or not the value has been changed from its time of display. This can either be statically set to "1" or you can update this variable when the data field changes to indicate to the web server module that the value should be written to the controller.
"t_@_value"	(text) The data value to be written to the controller tag.

The following example page displays a DINT tag from a controller and lets the user update the tag with a Submit button:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<script language="JavaScript">
</script>
</head>
<body >
<form action="/rokform/WriteLogixTags" method="POST">
TEST:
<input type="hidden"      name="redirect"      "value="/user/web/writeOneTag.asp">
<input type="hidden"      name="numtags"        value="1">
<input type="hidden"      name="t_1_tagname"     value="TEST">
<input type="hidden"      name="t_1_slot"        value="1">
<input type="hidden"      name="t_1_type"        value="DINT">
<input type="hidden"      name="t_1_display"     value="Decimal">
<input type="hidden"      name="t_1_changed"     value="1">
<input type="hidden"      name="t_1_value"       value="0">
<!--Read the current tag value -->
<font
color=<%ReadLogixTag("1,1","test_tag_string","STRING");%><%ReadLogixTag("1,1","TEST
","DINT");%></font>
<input type="submit" name="submit"      "value="Clear Alarms">
</form>
</body>
</html>
```


ACTION="/rokform/ReadLogixTag"

This method performs an instant data table read. The response is an HTML stream.

The form accepts these parameters:

Input:	Description:
path	path to the data, such as "1,1"
name	name of tag you want to read
	tag names are case sensitive
type	any of the following types: BOOL, SINT, INT, DINT, REAL or STRING
connected (optional)	for an unconnected request, enter one of the following: FALSE,false,F,f,0,NO,no,OFF,off for a connected request, enter one of the following: TRUE,true,T,t,1,YES,yes,ON,on

For example:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<script language="JavaScript">
function post()
{
document.getElementById('submitit').submit();
}
</script>
</head>
<body onload="post();">
<form action="/rokform/ReadLogixTag" id="submitit" method="POST">
<input type="hidden" name="name" value="test_tag_dint">
<input type="hidden" name="path" value="1,1">
<input type="hidden" name="type" value="DINT">
<input type="hidden" name="connected" value="true">
</form>
</body>
</html>
```

ACTION="/rokform/CIPMessage"

This method performs an instant CIP message request. The response is an HTML stream.

The form accepts these parameters:

Input:	Description:
path	path to the data, such as "1,1"
service	enter the CIP service name
class	enter the class ID number
instance (optional)	enter the instance number
attribute (optional)	enter the attribute number
member (optional)	enter the CIP member
data (optional)	size of the data
type	any of the following types: BOOL, SINT, INT, DINT, REAL or STRING
connected (optional)	for an unconnected request, enter one of the following: FALSE,false,F,f,0,NO,no,OFF,off for a connected request, enter one of the following: TRUE,true,T,t,1,YES,yes,ON,on

For example:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<script language="JavaScript">
</script>
</head>
<body >
<form action="/rokform/CIPMessage" method="POST">
<input type="hidden"      name="path"           "value="1,1">
<input type="hidden"      name="service"         value="5">
<input type="hidden"      name="class"           value="1">
<input type="hidden"      name="instance"        value="1">
<input type="hidden"      name="attribute"       value="">
<input type="hidden"      name="member"         value="">
<input type="hidden"      name="data"            value="">
<input type="hidden"      name="connected"      value="false">
<input type="submit"      name="submit"         value="RESET">
</form>
</body>
</html>
```

Notes:

Monitoring Diagnostics

How to Use This Chapter

The web server modules provide several different levels of diagnostics. There are user-oriented diagnostics, as well as more detailed diagnostics for technical support personnel. This chapter describes the diagnostics presented on the user-oriented diagnostic pages.

For this information:	See page:
Web Server Module Diagnostics	9-1
Diagnostics Overview	9-2
Network Settings	9-4
Message Connections	9-6
Ethernet Statistics	9-7

Web Server Module Diagnostics

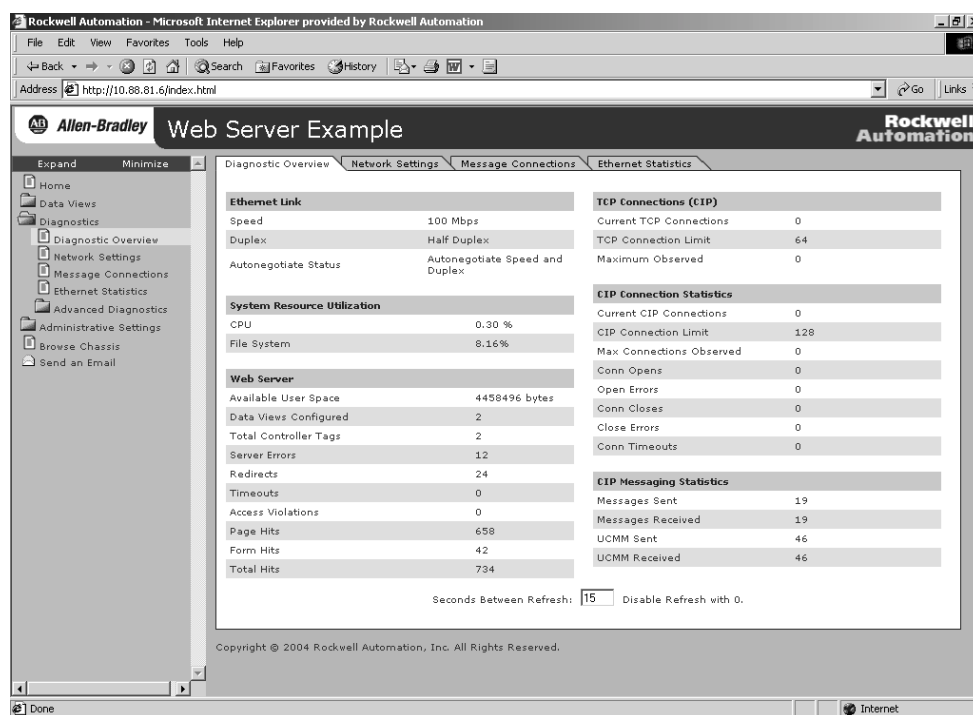
The web server module provides four diagnostic pages of user-oriented diagnostics. This information is organized into:

For this information:	Access this web page:
overview of the current configuration of the web server module	Diagnostics → Diagnostic Overview
summary of the network settings configured for the web server module	Diagnostics → Network Settings
statistics about messages initiated by the web server module and their associated connections	Diagnostics → Message Connections
Ethernet statistics	Diagnostics → Ethernet Statistics

Diagnostics Overview

The Diagnostics → Diagnostic Overview page presents a summary of the current configuration and overall status of the web server module. This summary includes:

- Ethernet configuration
- web server file use
- web server statistics
- TCP connection use
- CIP connection use
- message statistics



This field:	Specifies:
Ethernet Link	
Speed	whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex	whether the Ethernet port is operating at half duplex or full duplex
Autonegotiate Status	whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
System Resource Utilization	
CPU	current percent CPU utilization for the web server module
File System% Utilization	current percent utilization of the space available for user files

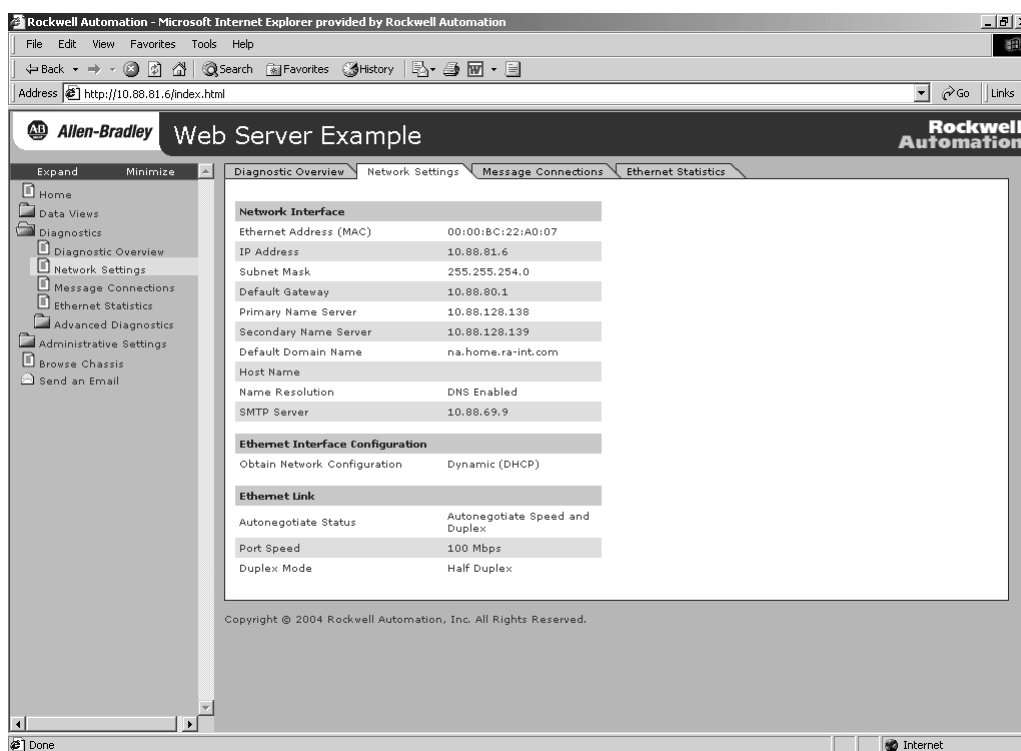
This field:	Specifies:
Web Server	
Available User Space	current number of bytes available for user files
Data Views Configured	total number of data views configured
Total Controller Tags	total number of tags in data views
Server Errors	number of requests to the web server module with an invalid URL
Redirects	number of requests for a web page that were redirected by the web server (e.g., requesting "/" is redirected to "/index.html")
Timeouts	number of times a connection timeout occurred while processing a web page
Access Violations	number of times a page has been requested for which the user has insufficient privilege
Page Hits	number of times a web page was successfully accessed
Form Hits	number of times a web page form was accessed
Total Hits	total number of web page access attempts
TCP Connections (CIP)	
Current TCP Connections	current number of active TCP connections for CIP messaging
TCP Connection Limit	maximum number of TCP connections for CIP messaging allowed
Maximum Observed	maximum observed number of TCP connections for CIP messaging
CIP Connection Statistics	
Current CIP Connections	current number of CIP connections
CIP Connection Limit	maximum number of CIP connections allowed
Max Connections Observed	maximum observed number of CIP connections
Conn Opens	number of CIP connection open requests
Open Errors	number of CIP connection open request errors
Conn Closes	number of CIP connection close requests
Close Errors	number of CIP connection close errors
Conn Timeouts	number of CIP connection timeouts
CIP Messaging Statistics	
Messages Sent	number of CIP connected messages sent
Messages Received	number of CIP connected messages received
UCMM Sent	number of CIP unconnected messages sent
UCMM Received	number of CIP unconnected messages received

CIP connections are for Logix-based communications, such as MSG instructions, RSLink communications, and PanelView communications. A CIP connection transfers data from one Logix application running on one end-node to a second Logix application running on another end-node. A CIP connection is established over a TCP connection.

Network Settings

The Diagnostics → Network Settings page presents a summary of the current Ethernet configuration for the web server module. This summary includes:

- Ethernet address details
- Ethernet interface type
- Ethernet network configuration



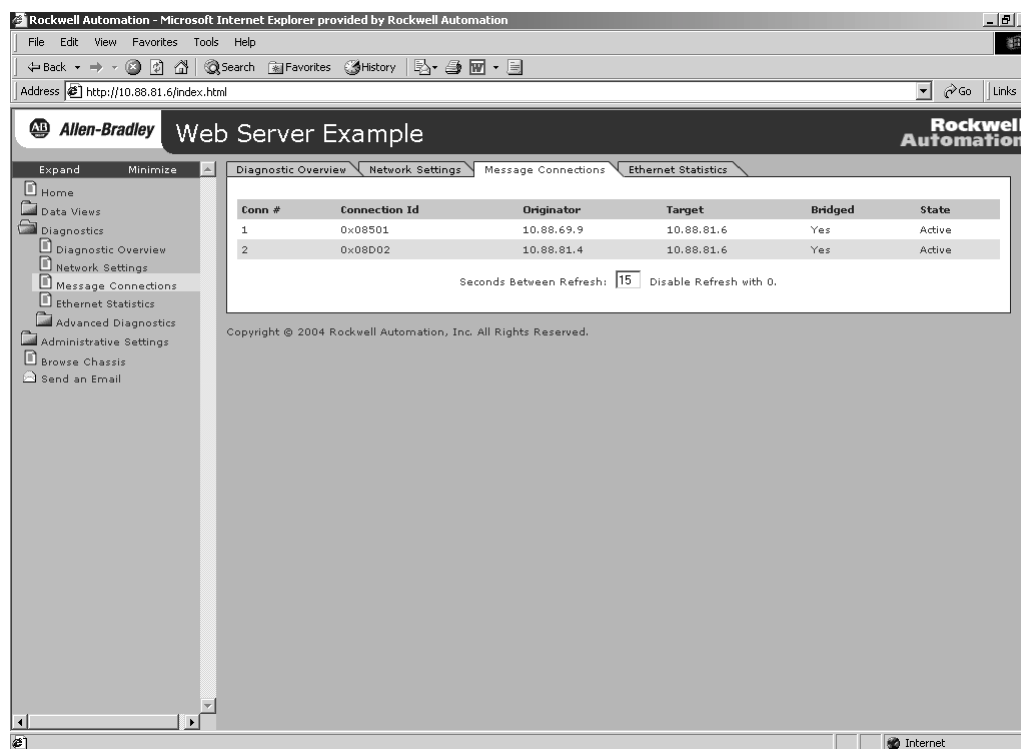
Any fields not configured remain blank:

This field:	Specifies:
Network Interface	
Ethernet Address (MAC)	Ethernet (MAC) address of the module
IP Address	IP address for the module
Subnet Mask	subnet mask for the module
Default Gateway	gateway address for the module
Primary Name Server	primary name server
Secondary Name Server	secondary name server
Default Domain Name	default domain name for the module
Host Name	host name for the module
Name Resolution	whether or not Domain Name System (DNS) resolution is enabled
SMTP Server	SMTP server address for the module (required for email).

This field:	Specifies:
Ethernet Interface Configuration	
Obtain Network Configuration	whether the module is configured to obtain its network parameters (IP address, etc.) via BOOTP, DHCP, or from static configuration
Ethernet Link	
Autonegotiate Status	whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Port Speed	whether the Ethernet port is operating at half duplex or full duplex
Duplex Mode	whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured

Message Connections

The Diagnostics → Message Connections page presents a summary of messages bridged through or initiated by the web server module.

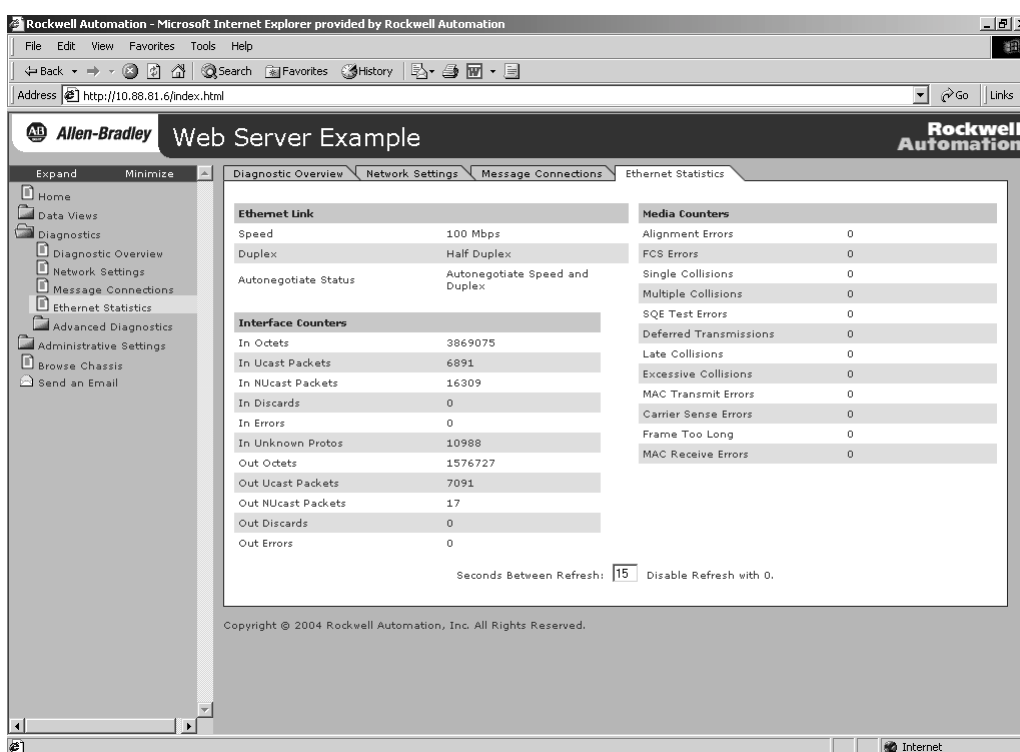


This field:	Specifies:
Conn #	the relative index of this connection (on the Message Connections page)
Connection ID	the unique identifier for each connection
Originator	the IP address of the device that originated the connection on Ethernet network
Target	the IP address of the device that is the target of the connection on Ethernet. This may not be the ultimate target of the connection (e.g., the target could be a Logix controller in a chassis).
Bridged	whether or not the connection bridges through the web server module
State	the current state of the connection: <ul style="list-style-type: none"> • active • closing • faulted • reserved

Ethernet Statistics

The Diagnostics → Ethernet Statistics page presents a summary of the status of communication activity on the Ethernet network. This summary includes:

- Ethernet network configuration
- packets sent and received over the Ethernet network
- frames sent and received over the Ethernet network



This field:	Specifies:
Ethernet Link	
Speed	whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex	whether the Ethernet port is operating at half duplex or full duplex
Autonegotiate Status	whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
Interface Counters	
In Octets	Octets received on the Ethernet interface
In Ucast Packets	Unicast packets received on the Ethernet interface
In NUCast Packets	Non-unicast packets received on the Ethernet interface
In Discards	Inbound packets received on the Ethernet interface but discarded
In Errors	Inbound packets that contain errors (does not include In Discards)
In Unknown Protos	Inbound packets with unknown protocol
Out Octets	Octets sent on the Ethernet interface

This field:	Specifies:
Out Ucast Packets	Unicast packets sent on the Ethernet interface
Out NUcast Packets	Non-unicast packets sent on the Ethernet interface
Out Discards	Outbound packets discarded
Out Errors	Outbound packets that contain errors
Media Counters	
Alignment Errors	Frames received that are not an integral number of octets in length
FCS Errors	Frames received that do not pass the FCS check
Single Collisions	Successfully transmitted frames which experienced exactly one collision
Multiple Collisions	Successfully transmitted frames which experienced more than one collision
SQE Test Errors	Number of times SQE test error message is generated
Deferred Transmissions	Frames for which first transmission attempt is delayed because the medium is busy
Late Collisions	Number of times a collision is detected later than 512 bit-times into the transmission of a packet
Excessive Collisions	Frames for which transmission fails due to excessive collisions
MAC Transmit Errors	Frames for which transmission fails due to an internal MAC sublayer transmit error
Carrier Sense Errors	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame
Frame Too Long	Frames received that exceed the maximum permitted frame size
MAC Receive Errors	Frames for which reception on the Ethernet interface failed due to an internal MAC sublayer receive error

Specifications

How to Use This Appendix

This appendix provides specifications and LED indications for the enhanced web server module.

For this information:	See page:
1756-EWEB Specifications	A-1
1756-EWEB LED Indicators	A-3

1756-EWEB Specifications

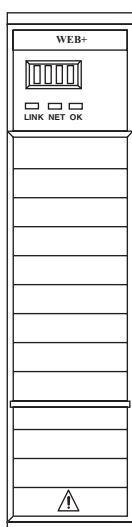
Specification:	Value:
TCP connections	64 TCP/IP connections
CIP connections	128 CIP connections, of which: <ul style="list-style-type: none"> • 128 bridged connections • 32 end-node connections 256 CIP unconnected messages (128 sending, 128 receiving)
Ethernet communication rate	10/100Mbps/s, half or full-duplex
Ethernet connector	RJ-45, category 5
Flash file memory	5 Mbytes flash memory 25 user accounts 2500 tag entries across all configured data views
maximum backplane current load	700mA @ 5.1V dc 3mA @ 24V dc from I/O chassis backplane
power dissipation	3.65W maximum
isolation voltage	50V
conductor category	wiring category - 2 (on communications ports) ⁽¹⁾
operating temperature	0 to 60°C (32 to 140°F)
IEC 60068-2-1 (Test Ad, Operating Cold)	
IEC 60068-2-2 (Test Bd, Operating Dry Heat)	
IEC 60068-2-14 (Test Nb, Operating Thermal Shock)	
storage temperature	-40 to 85°C (-40 to 185°F)
IEC 60068-2-1 (Test Ab, Un-packaged Non-operating Cold)	
IEC 60068-2-2 (Test Bb, Un-packaged Non-operating Dry Heat)	
IEC 60068-2-14 (Test Na, Un-packaged Non-operating Thermal Shock)	
relative humidity	5 to 95% non-condensing
IEC 60068-2-30 (Test Db, Un-packaged Non-operating Damp Heat)	
vibration (unpackaged)	2g @ 10-500Hz
IEC 60068-2-6 (Test Fc, Operating)	

A-2 Specifications

Specification:	Value:
operating shock IEC 60068-2-27 (Test Ea, Unpackaged Shock)	30g
non-operating shock IEC 60068-2-27 (Test Ea, Unpackaged Shock)	50g
emissions CISPR 11	Group 1, Class A
ESD Immunity IEC 61000-4-2	6kV contact discharges 8kV air discharges
radiated RF immunity IEC 61000-4-3	10V/m with 1kHz sine-wave 80%AM from 30MHz to 1000MHz 10V/m with 200Hz 50% Pulse 100%AM at 900Mhz
EFT/B immunity IEC 61000-4-4	±2kV at 5kHz on communications ports
surge transient immunity	±2kV line-earth(CM) on shielded ports
conducted RF immunity IEC 61000-4-6	10Vrms with 1kHz sine-wave 80%AM from 150kHz to 80MHz
certifications	UL: UL Listed Industrial Control Equipment CSA: CSA Certified Process Control Equipment CSA Certified Process Control Equipment for Class I, Division 2 Group A,B,C,D Hazardous Locations CE ⁽²⁾ : European Union 89/336/EEC EMC Directive, compliant with: EN 50081-2; Industrial Emissions EN 61326; Meas./Control/Lab., Industrial Requirements EN 61000-6-2; Industrial Immunity EN 61000-6-4; Industrial Emissions C-Tick ⁽²⁾ : Australian Radiocommunications Act, compliant with: AS/NZS CISPR 11; Industrial Emissions EtherNet/IP: ODVA conformance tested to EtherNet/IP specifications

⁽¹⁾ Use this Conductor Category information for planning conductor routing. See the *Industrial Automation Wiring and Grounding Guidelines*, publication 1770-4.1

⁽²⁾ See the Product Certification link at www.ab.com for Declarations of Conformity, Certificates, and other certification details.

1756-EWEB LED Indicators*Network (NET) status indicator*

State:	Status:	Description:
off	not powered, no IP address	Module is not powered, or does not have an IP address. <ul style="list-style-type: none"> • Verify there is chassis power and the module is completely inserted into the chassis and backplane. • Make sure the module has been configured.
flashing green	no CIP connections	Module has obtained an IP address, but has no established connections.
green	CIP connections	Module has an IP address and at least one established connection.
flashing red/green	self test	Module performing power-up self-test.

Link status indicator

State:	Status:	Description:
off	no data transmission	Module is not ready to communicate.
green	ready	Module is ready to communicate.
flashing green	data transmission in progress	Module is communicating over the network.

OK status indicator

State:	Status:	Description:
off	no power	Module does not have 24V DC power. Verify there is chassis power and the module is completely inserted into chassis and backplane.
flashing green	standby	Module does not have an IP address.
green	operational	Module is operating correctly.
flashing red	minor fault	A recoverable fault has been detected. This could be caused by an error in the configuration.
red	major fault	An unrecoverable fault has been detected. Recycle power to the module. If this does not clear the fault, replace the module.
flashing red/green	self test	Module performing power-up self-test.

Notes:

Connection Use by the Web Server Module Over Ethernet

How to Use This Chapter

A connection is a communication association between two devices. The web server module uses different types of communications connections. Since connections use internal module resources, the web server module has limits on the number of connections it supports. This appendix discusses these types of connections:

- CIP connections

The web server module uses CIP connections to read data view data, to communicate with a module in the local chassis, and to communicate with another EtherNet/IP module. For example, CIP connections are used when a Logix controller sends messages through the web server module to another Logix controller over Ethernet.

- TCP/IP connections for EtherNet/IP communications

As specified by the EtherNet/IP protocol, when the web server module communicates with another EtherNet/IP module, it first establishes a TCP connection with the target module. The TCP connection is then used for CIP connected and unconnected messages. Multiple CIP connections may be established over a single TCP connection.

For this information:	See page:
CIP Connections	B-2
TCP Connections	B-3

CIP Connections

Connections are allocations of resources that provide more reliable communications between modules than unconnected messages.

Examples of web-server functions that use CIP connections include:

- data views (one connection per controller slot)
- sending email initiated by controller MSG instructions
- bridging messages through the web server module

CIP communications take priority over HTTP communications. Generally, in the presence of heavy CIP traffic, HTTP traffic may appear “sluggish.”

There are different types of CIP connections:

CIP connection type:	Description:
bridged connection	A bridged connection is a connection that passes through the web server module. The end point of the connection is a module other than the web server module. Example: a connection from a controller through a 1756-ENBT to another controller.
end-node connection	An end-node connection is a connection whose end point is the web server module itself. Example: a connection from RSLinx to the web server module to set the IP address.

You cannot bridge I/O or produced/consumed tags through the web server module.

CIP connected messaging limits

Product:	CIP Connected Messaging Limits:
1756-EWEB	Each module supports: <ul style="list-style-type: none"> • 128 connections for any combination of data views, bridged messages, and ASP function calls • 32 connections can be end-node connections

CIP unconnected messaging limits

The following limits of unconnected messages are the maximum number of outstanding unconnected messages. These are unconnected messages that are sent to the or bridged through the web server module.

Product:	CIP Unconnected Messaging Limits:
1756-EWEB	<p>Each module supports:</p> <ul style="list-style-type: none"> • 256 CIP unconnected messages, of which: <ul style="list-style-type: none"> 128 can be unconnected messages from the Ethernet port to an object on the module or to the backplane. 128 can be unconnected messages from the backplane to an object on the module or to the Ethernet port.

See the *EtherNet/IP Performance and Application Guide*, publication number ENET-AP001, for more information on connections.

TCP Connections

The web server module uses one TCP connection for each IP address to which the module is connected. Multiple CIP connections can go through a single TCP connection. Examples of TCP connections are:

- HMI (human-machine interface) to a controller that supports EtherNet/IP communications
- Logix MSG instruction to a controller or workstation
- OPC or DDE accessing a controller

The web server module supports 64 TCP connections that handle CIP connections.

Notes:

Numerics**1756-EWEB** 1-1, A-1, A-3**A****access levels**

classes 6-1

access limits

configuring 6-2

pre-defined pages 6-3

accessing

custom web pages 8-3

data views 1-6

Home page 1-3, 1-4

typical application 1-1

Administrator access 6-2**applications** 1-1**ASP**

CIPMessage 8-5

CIPMessageUnconnected 8-6

custom web pages 8-1

developing 8-3

enabling 1-9

function calls 8-4

GetSetting 8-7

read CIP data 8-5

read controller tags 8-4

ReadLogixTag 8-4

ReadLogixTagUnconnected 8-5

retrieve web server information 8-7

authentication 6-1**B****backing up**

file system 7-5

procedure 7-6

BOOTP/DHCP utility 2-6**bridging** 1-2**browser requirements** 1-3**C****CIP connections** B-2**CIPMessage** 8-5, 8-20**CIPMessageUnconnected** 8-6**configuring**

access limits 6-2

device services 3-4

DNS addressing 2-10

email 1-7, 5-3

Home page 3-2

module information 3-1

MSG instructions 5-7

network parameters 2-1, 3-3

server log 3-8

server settings 3-5

services 1-9

time server 1-8

timer server 3-7

user accounts 6-5

configurng

DHCP server 2-3

connecting 1-1, 7-2**connections**

CIP B-2

overview B-1

TCP' TCP connections B-3

control system data 8-7**creating**

data views 1-5, 4-3, 4-8

offline 4-8

custom web pages

accessing 8-3

ASP function calls 8-4

CIPMessage 8-5, 8-20

CIPMessageUnconnected 8-6

conversion.js 8-9

data views 8-13

developing 8-3

forms 8-12

GetSetting 8-7

Javascript libraries 8-9

overview 8-1

POST handlers 8-12

read CIP data 8-5

read controller tags 8-4

ReadLogixTag 8-4

ReadLogixTags 8-19

ReadLogixTagUnconnected 8-5

retrieve web server information 8-7

updating control system data 8-7

WriteLogixTags 8-17

XMLObjectLoaderLib.js 8-10

D

data views

- /user/system/dataviews 4-8
- accessing 1-6
- adding tags 4-4
- creating 1-5, 4-3
- editing 4-7
- examples 4-10, 4-11, 4-12
- external application 4-9
- interfacing with Logix5000 controller 4-6
- monitoring 4-5
- offline 4-8
- overview 4-1
- performance estimates 4-2
- post tags 8-13
- sorting 4-5
- supported tags 4-2

device configuration 3-1

device services 1-9, 3-4

DHCP server 2-3

diagnostics

- diagnostics overview 9-2
- Ethernet statistics 9-7
- message connections 9-6
- network settings 9-4
- web server module 9-1

directories 7-1

DNS addressing 2-10

E

editing

- data views 4-7

email

- configuring 1-7
- configuring web server 5-3
- MSG instruction 5-7
- overview 5-1
- sending via MSG instruction 5-4
- sending via web page 5-4
- status codes 5-10
- text format 5-9
- typical application 1-2

example data view 4-10, 4-11, 4-12

external applications 4-9

F

file names 7-3

file system 7-5

file types 7-3

forms 8-12

FTP

- enabling 1-9

ftp

- /user/web 7-3
- backing up procedure 7-6
- backing up the file system 7-5
- connecting 7-2
- directories 7-1
- file names 7-3
- file types 7-3
- restoring files 7-7

function calls 8-4

G

GetSetting 8-7

getting started 1-1

H

Home page 1-3, 1-4, 3-2

I

installing 1-1

IP address 1-2, 2-1, 2-4

J

Javascript libraries

- conversion.js 8-9
- overview 8-9
- XMLObjectLoaderLib.js 8-10

L

LEDs A-3

log 3-8

logging on 1-3

Logix5000 controllers 4-6

M

module information

- device configuration 3-1

MSG instructions 5-7

N

navigating 1-4

network configuration

- confirming 1-4

network parameters

- BOOTP/DHCP utility 2-6
- configuring 2-1
- DNS addressing 2-10
- modifying 3-3
- RSLinux software 2-8
- RSLogix 5000 software 2-9
- verifying 2-12
- with DHCP server 2-3
- without DHCP server 2-5

P

- password** 6-1, 6-6
- POST handlers** 8-12
- pre-defined pages** 6-3

Q

- quick start** 1-1

R

- Read access** 6-2
- read CIP data** 8-5
- read controller tags** 8-4
- ReadLogixTag** 8-4
- ReadLogixTags** 8-19
- ReadLogixTagUnconnected** 8-5
- recovering** 6-6
- remote access** 1-1
- requirements, browser** 1-3
- restoring files** 7-7
- retrieve web server information** 8-7
- routing** 1-2
- RSLinux software** 2-8
- RSLogix 5000 software** 2-9

S

- server log** 3-8
- server settings** 3-5, 3-6
- services** 1-9
- SMTP** 1-9
- SNMP** 1-9
- sorting data views** 4-5
- specifications** A-1
- status codes, email** 5-10
- string tags** 5-5

T

- tags** 4-2
- time server** 1-8
- timer server** 3-7
- typical applications** 1-1

U

- updating control system data** 8-7
- user accounts**
 - classes 6-1
 - creating 6-5

V

- verifying, network parameters** 2-12

W

- Write access** 6-2
- WriteLogixTags** 8-17

X**XML**

- data views 4-1
- enabling 1-9
- example data view 4-10, 4-11, 4-12
- format 4-5
- typical application 1-2

Notes:

ControlLogix and RSLogix are trademarks of Rockwell Automation.

Ethernet is a trademark of Digital Equipment Corporation, Intel and Xerox.

EtherNet/IP is a registered trademark of Digital Equipment Corporation, Intel and Xerox.

This product incorporates code from GoAhead Software, Inc.,
Copyright © 2004 GoAhead Software, Inc. All Rights Reserved.



How Are We Doing?

Your comments on our technical publications will help us serve you better in the future.
Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at
RADocumentComments@ra.rockwell.com

Pub. Title/Type Ethernet Web Server Module User Manual

Cat. No. 1756-EWEB Pub. No. ENET-UM527A-EN-P Pub. Date March 2004 Part No. 957859-77

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness 1 2 3	How can we make this publication more useful for you?		
Completeness (all necessary information is provided) 1 2 3	Can we add more information to help you?		
	procedure/step	illustration	feature
	example	guideline	other
	explanation	definition	
Technical Accuracy (all provided information is correct) 1 2 3	Can we be more accurate?		
	text	illustration	
Clarity (all provided information is easy to understand) 1 2 3	How can we make things clearer?		
Other Comments	You can add additional comments on the back of this form.		

Your Name _____
Your Title/Function _____
Location/Phone _____

Would you like us to contact you regarding your comments?
___ No, there is no need to contact me
___ Yes, please call me
___ Yes, please email me at _____
___ Yes, please contact me via _____

Return this form to: Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705
Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE



**Rockwell
Automation**

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705



Rockwell Automation Support

Rockwell Automation provides technical information on the web to assist you in using our products. At <http://support.rockwellautomation.com>, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration and troubleshooting, we offer TechConnect Support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit <http://support.rockwellautomation.com>.

Installation Assistance

If you experience a problem with a hardware module within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your module up and running:

United States	1.440.646.3223 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

New Product Satisfaction Return

Rockwell tests all of our products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned:

United States	Contact your distributor. You must provide a Customer Support case number (see phone number above to obtain one) to your distributor in order to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for return procedure.

www.rockwellautomation.com

Corporate Headquarters

Rockwell Automation, 777 East Wisconsin Avenue, Suite 1400, Milwaukee, WI, 53202-5302 USA, Tel: (1) 414.212.5200, Fax: (1) 414.212.5201

Headquarters for Allen-Bradley Products, Rockwell Software Products and Global Manufacturing Solutions

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europe: Rockwell Automation SA/NV, Vorstlaan/Boulevard du Souverain 36-BP 3A/B, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia Pacific: Rockwell Automation, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Headquarters for Dodge and Reliance Electric Products

Americas: Rockwell Automation, 6040 Ponders Court, Greenville, SC 29615-4617 USA, Tel: (1) 864.297.4800, Fax: (1) 864.281.2433

Europe: Rockwell Automation, Brühlstraße 22, D-74834 Elztal-Dallau, Germany, Tel: (49) 6261 9410, Fax: (49) 6261 17741

Asia Pacific: Rockwell Automation, 55 Newton Road, #11-01/02 Revenue House, Singapore 307987, Tel: (65) 351 6723, Fax: (65) 355 1733



Allen-Bradley

EtherNet/IP Web Server Module

User Manual